Année 2004
N° d'ordre : 752

THÈSE

Préparée au
Laboratoire d'Analyse et d'Architecture des Systèmes du CNRS

En vue de l'obtention du titre de
Docteur de l'Institut National des Sciences Appliquées de Toulouse

Spécialité
Systèmes Industriels

Par
Tatiana KEMPOWSKY

SURVEILLANCE DE PROCEDES A BASE DE METHODES DE CLASSIFICATION : CONCEPTION D'UN OUTIL D'AIDE POUR LA DETECTION ET LE DIAGNOSTIC DES DEFAILLANCES

Soutenue le 14 décembre 2004, devant le jury :

Rapporteurs :	J-P. CASSAR S. GENTIL
Directeurs de thèse :	J. AGUILAR-MARTIN A. SUBIAS
Examinateurs :	F. JIMENEZ A. TITLI
Invité(s) :	N. AGELL M-V. LE LANN

Rapport LAAS N°

Remerciements

*Les travaux présentés dans ce mémoire ont été réalisés au Laboratoire d'Analyse et d'Architecture des Systèmes du Centre National de la Recherche Scientifique (LAAS-CNRS) au sein du groupe Diagnostic, Supervision et Conduite qualitatifs (DISCO). Je tiens tout d'abord à remercier les directeurs successifs du LAAS, Messieurs **Jean-Claude Laprie** et **Malik Ghallab** pour leur accueil.*

*Je tiens à remercier Monsieur **Joseph Aguilar-Martin** et Madame **Louise Travé-Massuyès**, successivement responsables du groupe DISCO, de m'y avoir accueilli pendant ces années de thèse.*

*J'adresse toute ma gratitude à mes directeurs de thèse Madame **Audine Subias** et Monsieur **Joseph Aguilar-Martin** pour leur collaboration inestimable, leur disponibilité et les nombreuses conversations et conseils qui m'ont aidée et orientée non seulement dans mon travail mais aussi dans la vie quotidienne. Je voudrais les remercier aussi pour toute la confiance qu'ils ont eu en moi, pour leur patience et leur soutien.*

*Je suis très reconnaissante de l'honneur que m'ont fait Madame **Sylviane Gentil**, Professeur à l'INP de Grenoble et Monsieur **Jean-Philippe Cassar**, Professeur à l'Université de Lille 1, en acceptant la tâche d'évaluer en qualité de rapporteur les travaux présentés. J'exprime également ma reconnaissance à Messieurs **André Titli**, Professeur à l'INSA de Toulouse, **Fernando Jimenez**, Professeur à l'Université des Andes, Colombie et Madame **Nuria Agell**, Professeur à l'Universitat Ramon Llull, Barcelona, pour avoir accepté de prendre part au jury. Je remercie également à Madame **Marie-Véronique Le Lann**, Professeur à l'INSA de Toulouse, pour ses remarques, ses conseils et son soutien et en fin pour avoir accepté de prendre part au jury. Je leur remercie tous pour l'intérêt qu'ils ont porté à ces travaux.*

Une grande partie de la thèse a été financée par le projet européen CHEM. Je voudrais remercier les partenaires du projet et toutes les personnes qui m'ont apporté leur collaboration.
*Je tiens à remercier tout particulièrement toutes les personnes qui ont contribué au développement de l'outil SALSA. Je tiens à exprimer ma gratitude à **Bernard Franc** pour sa disponibilité et son aide chaque fois que j'ai rencontré un problème informatique.*

*Durant ces années passées au LAAS, et à Toulouse, j'ai pu apprécier l'amitié de certaines personnes qui, au-delà de l'ambiance très amicale qu'elles ont créée, m'ont toujours apporté leur soutien dans les moments critiques. Je voudrais remercier très sincèrement **Emmanuelle Despontin** pour ses encouragements et son amitié (je n'ai pas assez de mots pour lui exprimer ma gratitude). Je souhaiterais aussi remercier mes collègues du groupe DISCO en particulier **Miguel Gonzalez** et **Victor-Hugo Grisales** qui ont toujours été prêts à répondre à mes interrogations de nature très diverse. A mes colocataires de bureau, **Antonio** et **Hector** je dis merci pour tout ce que nous avons pu partager.*

*Mes remerciements à tout le personnel technique et administratif du LAAS, avec une attention particulière envers **Eliane Dufour** et **Christian Berty** pour leur amabilité, leur efficacité et leur temps.*

*Enfin, il y a des choses que ne peuvent pas être remerciées par écrit, néanmoins je voudrais remercier **Juan-Carlos** d'avoir été toujours à mes cotés dans cette aventure, de croire en moi plus que personne d'autre, de ne pas m'avoir laissé baisser les bras dans les moments les plus difficiles et bien sûr d'avoir partagé aussi les moments les plus heureux. A ma **famille**, pour leur soutien et leur compréhension : là où je serai je sais qu'elle sera toujours près de moi.*

Table des matières

Liste des figures

INTRODUCTION GENERALE

Les entreprises de nombreux secteurs d'activité (automobile, sidérurgie, chimie, pétro-chimie, …) sont de plus en plus soumises à la concurrence du marché. Pour assurer leur avenir, les entreprises doivent désormais faire face à différents enjeux socio-économiques, ce qui les amène à avoir une complexité toujours croissante de leurs systèmes de production pour pouvoir atteindre des objectifs de plus en plus exigeants. Sur le plan économique, les coûts de production, le rendement ainsi que le respect des délais sont des facteurs qui influent sur la compétitivité des entreprises. Sur le plan technique, les principales contraintes portent sur la diversité, la flexibilité, la complexité et la qualité des produits. Des développements au niveau des technologies de l'informatique et de l'automatisation deviennent nécessaires.

L'automatisation est donc devenue de plus en plus importante pour optimiser la productivité en améliorant la maîtrise des processus industriels. Malheureusement, pour atteindre ses objectifs, l'automatisation a accru la complexité des processus et a changé la tâche de l'opérateur dont le rôle a évolué de la conduite à la supervision. Désormais, il est devenu essentiel que l'opérateur puisse connaître, à tout instant, l'état de fonctionnement du processus. Pour cela, l'échange d'informations entre l'opérateur et le processus a été amélioré au travers des *interfaces home-machine*. Mais au-delà de ces évolutions, il faut désormais être capable aussi de détecter un dysfonctionnement le plus rapidement possible, de l'isoler, d'en identifier la cause probable de façon à réduire leurs conséquences néfastes, et puis de proposer à l'opérateur des actions correctives.

Des nos jours, l'implémentation des systèmes automatisés suppose donc la mise en place d'outils pour la supervision. La maîtrise d'un processus est indissociable de sa supervision pour aider les entreprises dans leur recherche permanente d'une meilleure productivité et qualité à moindre coût. Cette supervision permet, par ailleurs, aux entreprises de garantir et préserver la sûreté de fonctionnement des équipements, de leur personnel (opérateurs), mais aussi de protéger l'environnement.

Dans ce contexte, de nombreuses approches sont développées, en vue de la détection de défaillances et du diagnostic, par les différentes communautés de recherche en automatique, productique et intelligence artificielle. Les méthodes se différencient par rapport au type de connaissances a priori sur le processus qu'elles nécessitent. Ainsi elles peuvent être classées, de façon générale, comme des méthodes à base de modèles, à base de connaissances et des méthodes à base de données historiques. Les méthodes à base de modèles considèrent un modèle structurel du comportement du processus basé sur des principes physiques fondamentaux. Ces modèles peuvent être de type quantitatif, exprimés sous forme d'équations mathématiques ou bien de type qualitatif, exprimés par exemple sous forme de relations logiques. Les méthodes à base de connaissance exploitent les compétences, le raisonnement et les connaissances des experts sur le processus pour les transformer en règles, de manière à résoudre des problèmes spécifiques. Enfin, les méthodes à base de données cherchent à découvrir des informations, sous forme d'exemples type ou tendances, au sein des mesures venant des capteurs et des actionneurs, pouvant identifier le comportement du procédé. Ces méthodes comprennent, parmi d'autres, les méthodes d'apprentissage et de classification (ou reconnaissance de formes).

Les travaux développés par le groupe DISCO (DIagnostic, Supervision et COnduite) au sein du LAAS (Laboratoire d'Analyse et d'Architecture des Systèmes) se situent dans ce cadre de la maîtrise des systèmes complexes. Les thèmes de recherche abordés couvrent les aspects : commande (régulation, poursuite) supervision, surveillance, diagnostic, aide à la conduite et interaction homme-système. Les travaux développés s'appuient sur trois types d'approches que sont les approches à base de modèles, à base de connaissances et par apprentissage et reconnaissance de formes. En ce qui concerne les approches par apprentissage et reconnaissance des formes, des travaux récents ont été menés au niveau de la supervision et de la surveillance des systèmes complexes. Ces travaux, plus particulièrement axés sur les bio-procédés et les processus de traitement des eaux, ont abordé les problèmes liés à l'interprétation des données (*Data Mining*) et à l'extraction des informations à partir d'une méthode de classification avec apprentissage [WAI00].

Notre travail se situe dans un contexte voisin puisqu'il porte sur la supervision et la surveillance de procédés complexes, tels que les procédés chimiques et pétrochimiques.

Pour ce type de procédé, l'obtention d'un modèle mathématique n'est pas toujours possible. Notre contribution a donc consisté à proposer une stratégie permettant, à partir des données disponibles et à l'aide d'une méthode de classification, de construire un modèle du comportement du processus. La mise en place de ce modèle est réalisée avec l'objectif de suivre en ligne les évolutions du procédé surveillé, d'identifier des situations anormales et d'aider l'opérateur dans la prise de décisions, surtout lorsque ces situations anormales ont pour origine des dysfonctionnements de la partie opérative.

Ce mémoire de thèse est structuré en cinq chapitres :

Le premier chapitre propose un tour d'horizon sur la supervision et les différentes approches de détection et de diagnostic de défaillances développées par les différentes communautés de recherche. De plus, une tentative d'intégration des différentes méthodes, pour la supervision et l'aide à la décision des opérateurs des procédés complexes, est abordée. Cette tentative d'intégration et de collaboration entre différentes méthodes a été l'objectif d'un projet international pour lequel une partie importante de la thèse a été développée.

Le deuxième chapitre propose une stratégie pour le suivi du comportement d'un processus et la détection de défaillances, en s'appuyant sur des données historiques et des données en ligne. Ce chapitre décrit, également, différentes méthodes de classification et d'apprentissage qui peuvent être utilisées pour la mise en place de la stratégie proposée, dont la méthode LAMDA (Learning Algorithm for Multi-variable Data Analysis) sur laquelle nous avons appuyé nos travaux.

Dans le troisième chapitre, nous abordons les différentes étapes à suivre pour l'élaboration d'un système de surveillance à partir de méthodes de classification. Ainsi, la première phase consiste à réaliser un apprentissage pour identifier et caractériser les différents états de fonctionnement du processus à surveiller. Une description des éléments de paramétrage nécessaires pour l'élaboration du modèle de comportement du processus est aussi réalisée. La non exhaustivité du modèle de comportement élaboré à partir des données met en évidence la nécessité d'avoir un système de surveillance adaptatif. Ainsi, la deuxième phase que nous présentons consiste à faire une reconnaissance en ligne des situations connues et à suivre une démarche précise dans les cas de détection de déviations de comportement. Finalement, dans ce chapitre, nous présentons l'outil SALSA qui a été développé pour mettre en œuvre la stratégie proposée dans le chapitre 2. Cet outil offre une interface modulaire, flexible et conviviale, qui permet l'interaction et le dialogue avec l'expert, dans la construction du modèle de comportement. SALSA apporte, aussi, une aide à l'opérateur au moment du suivi et lors de la détection des déviations.

Le chapitre 4 est dédié à la mise en place d'un modèle d'évolutions des états fonctionnels du système surveillé. Le modèle est élaboré sous la forme d'un automate à états finis. Nous décrivons, dans cette partie, les moyens utilisés pour obtenir les états, les transitions et les événements associés à ces transitions. Nous montrons, également, comment cet automate peut être utilisé par l'opérateur, à des fins de surveillance.

Nous terminons ce mémoire avec un chapitre 5 dédié aux résultats obtenus lors de l'application de l'outil SALSA dans le domaine industriel. Trois unités pilotes ont été choisies pour illustrer notre travail.

Chapitre 1

1. SUPERVISION ET AIDE A LA DECISION DES SYSTEMES COMPLEXES

1.1 Introduction

Les systèmes industriels sont devenus de plus en plus complexes, pas seulement à cause de la complexité physique des installations et des différentes procédures pour la fabrication des produits de qualité, mais aussi de par la croissante quantité d'information qu'il est nécessaire de gérer pour en garantir le fonctionnement correct. La commande automatisée de ces systèmes a été enrichie progressivement grâce aux avances technologiques dans tous les domaines : les capteurs, les régulateurs, les réseaux à haute vitesse, etc. De plus, il est possible d'établir des liaisons de communication plus rapides, de récupérer et stocker chaque fois davantage d'information. Et même si les consoles des salles de contrôle sont plus performantes et plus versatiles, la quantité d'information affichée peut devenir ingérable.

Malgré le progrès et les avances technologiques dans la commande distribuée, les systèmes prédictifs et l'automatisation des procédés complexes, la supervision de processus reste une tâche très importante qui est encore en grande partie une activité manuelle, exécutée par les opérateurs, notamment lorsqu'il s'agit de répondre aux événements anormaux. Cette activité, peut selon la nature et la criticité des anomalies ou dysfonctionnements, avoir un impact économique, environnemental et de sécurité plus ou moins significatif, non seulement au niveau des équipements mais aussi des opérateurs. D'autant plus que le fait de gérer une grande quantité d'information et d'avoir besoin d'agir vite peut mener les opérateurs à prendre des décisions incorrectes, dégradant encore plus la situation.

De ce fait, les systèmes de supervision qui intègrent les outils de surveillance, de détection et de diagnostic sont nécessaires afin de fournir à l'opérateur des critères suffisants pour la prise des décisions.

Notre objectif, dans ce premier chapitre, est de faire un tour d'horizon des différentes techniques utilisées pour la supervision, le diagnostic et l'aide à la décision des processus industriels. Ceci va nous permettre d'identifier leurs champs d'application potentiels, leurs points forts et leurs faiblesses afin d'utiliser la technique ou l'ensemble des techniques appropriées, selon les cas. Après ce bilan, nous présenterons un projet international dont l'objectif justement était de proposer l'intégration et la combinaison de méthodes et d'outils adaptés à des contraintes techniques, économiques, environnementales et à des exigences de sécurité bien identifiées dans des systèmes industriels complexes tels que les procédés chimiques et pétrochimiques.

1.2 Supervision des procédés

Comme nous l'avons évoqué, le nombre des mesures recueillies sur les processus et stockées a augmenté de façon considérable. En même temps, ces processus ont accru leur complexité et les systèmes de commande ont remplacé les ajustements manuels. Ceci afin de répondre aux demandes de qualité, aux contraintes environnementales, à la réduction des coûts de production et maintenance, tout en garantissant la sécurité des installations ainsi que celle des opérateurs humains. Face à cet incrément d'information, changeante et dynamique, l'opérateur humain a besoin de nouveaux outils qui l'aident dans sa tâche de surveillance du processus, pour en garantir le fonctionnement correct et réagir au cas où des événements anormaux se produisent.

Dans ces conditions, nous pouvons dire que la supervision a pour objectif d'assurer la gestion réactive et sûre des modes de fonctionnement d'un processus. Ces modes ou situations sont définis à partir de l'***analyse*** des données, de la ***connaissance*** du système et du ***savoir-faire*** des opérateurs. Il faut alors exploiter au maximum toutes les informations disponibles sur le système pour pouvoir détecter les éventuels dysfonctionnements d'un processus et les diagnostiquer et réagir en conséquence de façon à assurer son fonctionnement même en situations anormales.

Au niveau de la supervision, toute description du procédé, qui apporte une connaissance *a priori* sur ses caractéristiques et ses fonctionnalités, constitue un ***modèle*** du procédé. Ceci permet de comparer l'évolution du procédé réel au travers du suivi des mesures à la description théorique offerte par le modèle. Le résultat de cette comparaison détecte le bon ou mauvais fonctionnement du procédé. Nous pouvons considérer le modèle comme la façon de valider le fonctionnement correct du procédé et de déterminer les ***déviations*** par rapport aux conditions attendues d'opération. Les modèles peuvent être de différentes natures selon les informations disponibles sur le processus : il existe des modèles de type analytique (équations différentielles, équations aux différences, relations entre variables, etc.), ainsi que des modèles qualitatifs (équations qualitatives, modèles à base d'ensembles flous, règles, description du comportement, etc.), qui représentent le fonctionnement statique ou dynamique, normal ou anormal du procédé.

Pour la mise en place d'un système de supervision, du point de vue de la communauté du continu, trois fonctions doivent être prises en compte : la ***détection***, le ***diagnostic*** et la ***reconfiguration***. Selon [KOT93], la ***surveillance*** du procédé traite les données disponibles en ligne, afin d'obtenir son état de fonctionnement. Dans la surveillance, nous retrouvons les fonctions de détection de défaillances et de diagnostic (Figure 1.1). De façon générale, la **détection** consiste en l'identification des changements ou déviations des mesures du procédé par rapport au fonctionnement normal, ce qui se traduit par la génération des ***symptômes***. Le **diagnostic** consiste à déterminer quelles sont l'origine et/ou la (les) cause(s) qui ont pu engendrer le symptôme détecté. A ce stade, le système doit avoir la capacité de décider quand le procédé se trouve dans une situation de fonctionnement normal, et quand une action corrective doit être appliquée. Cette action corrective correspond à l'étape de **reconfiguration** de la commande de façon à ramener le procédé dans un mode de fonctionnement normal.

Cependant, il existe d'autres approches pour la mise en place d'un système de supervision. En effet, pour la communauté de systèmes à événements discrets (SED), la **supervision** a pour but de contrôler l'exécution d'une opération effectuée par le système de commande sans rentrer dans les détails de cette exécution. La supervision a lieu dans une structure hiérarchique (au moins avec 2 niveaux), et recouvre l'aspect du fonctionnement normal et anormal :

- en fonctionnement normal, le rôle de la supervision est de prendre, en temps réel, les décisions correspondantes aux degrés de liberté exigés par la flexibilité décisionnelle;
- en présence de défaillances, la supervision prend toutes les décisions nécessaires pour le retour vers un fonctionnement normal. Après avoir déterminé un nouveau mode de fonctionnement, il peut s'agir de choisir une solution curative, d'effectuer des ré-ordonnancements "locaux", ou même de déclencher des procédures d'urgence.

La **surveillance** est responsable de l'acquisition des signaux en provenance des ressources et de la commande. Ces informations sont utilisées pour la reconstitution de l'état réel du système commandé et pour faire les inférences nécessaires afin de produire des informations supplémentaires pour dresser des historiques de fonctionnement. Les activités de la surveillance sont donc limitées aux fonctions relatives aux informations et n'ont pas une action directe sur le modèle ni sur le procédé. A priori, la surveillance a un rôle passif vis-à-vis de la commande. Parmi les fonctions de la surveillance nous trouvons donc, en plus de l'acquisition de données, la **détection** [COM00] qui caractérise le fonctionnement du système de normal ou anormal. Deux classes d'anomalies sont distinguées :

- la première regroupe les situations pour lesquelles le comportement du système devient anormal car les contraintes d'opération ne sont pas garanties;

- la deuxième regroupe les situations dans lesquelles le comportement est anormal par rapport à la loi de commande appliquée. Cette classe recouvre les anomalies de fabrication mise en évidence par des contrôles de qualité.

Le **suivi** fait partie aussi des fonctions de la surveillance. Cette fonction maintient en permanence un historique des traitements effectués, et une trace des événements que perçoit le système. La fonction **diagnostic** établi un lien de cause à effet entre un ***symptôme*** observé et la défaillance qui est survenue, ses causes et ses conséquences. La communauté des SED distingue trois sous-fonctions [COM00]:

- la localisation, qui détermine le sous-système responsable de la défaillance,
- l'identification, qui détermine les causes qui ont engendré la défaillance,
- l'explication, qui justifie les conclusions du diagnostic.

La fonction **pronostic** est également une fonction de surveillance qui a pour but de déterminer les conséquences d'une défaillance sur le fonctionnement futur du système. Il existe aussi des fonctions propres à la supervision :

- la **reconfiguration** qui agit sur le procédé en adaptant la configuration matérielle à la situation, ainsi que sur le système de commande en changeant la loi de commande,
- la **décision**, qui détermine l'état à atteindre pour le retour en opération normale et la séquence d'actions correctives à réaliser pour arriver à cet état.

Dans ces conditions, la supervision n'est plus un simple enchaînement des fonctions de surveillance (détection - diagnostic) et reconfiguration. Des "modèles de surveillance" beaucoup plus complexes peuvent être mis en place en fonction du procédé considéré, du type de défaillances et également de la politique de production de l'entreprise [ZAM98].

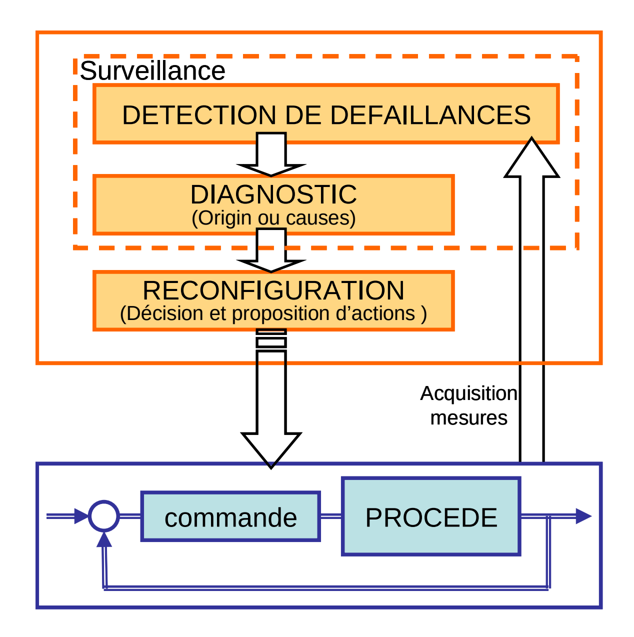


Figure 1.1. Schéma général de supervision

Nous allons focaliser notre travail sur la supervision des systèmes industriels et, plus particulièrement, sur les fonctions de surveillance telles que définies par le point de vue de la communauté du continu, (voir pointillé Figure 1.1) c'est-à-dire la détection et le diagnostic.

Différentes méthodes peuvent être utilisées pour la mise en œuvre des étapes de la supervision. Ainsi, nous pouvons classer les diverses méthodes utilisées pour détecter les anomalies et les diagnostiquer d'après la nature de la connaissance disponible sur le processus.

1.3 Différentes méthodes pour l'élaboration d'un système de supervision et diagnostic industriel

A l'heure actuelle, il existe plusieurs communautés de recherche, telles que la communauté de l'automatique, de la productique, de l'intelligence artificielle, qui s'intéressent au domaine de la supervision, et plus particulièrement, au diagnostic, dans le but principal d'assister les opérateurs dans la gestion des événements anormaux. La littérature sur le diagnostic de défaillances dans les processus industriels est vaste, s'étendant dès méthodes analytiques à l'intelligence artificielle et aux méthodes statistiques. D'un point de vue de la modélisation, des méthodes de diagnostic ont besoin de modèles précis du processus, de modèles semi-quantitatifs ou bien de modèles qualitatifs. Par ailleurs, d'autres méthodes n'utilisent aucune forme de modèle, et s'appuient seulement sur des données historiques du processus. Pour la description de ces différentes méthodes de diagnostic, nous avons adopté la classification proposée par Vankatasubramanian et al. [VEN03a]. Cette classification a été faite selon la connaissance du processus disponible. Nous précisons que cet état de l'art n'est pas exhaustif car notre objectif est uniquement de situer ces méthodes par rapport à leurs applications potentielles. D'autres façons de classer les diverses méthodes et techniques de diagnostic ont été proposées [ZWI95].

1.3.1 Méthodes à base de modèles

Les méthodes de diagnostic à base de modèles (connues aussi comme *diagnostic à partir de principes premiers*) [REI92] sont appropriées quand l'information disponible sur le processus permet d'utiliser les principes physiques du processus et leur compréhension. Selon la connaissance du processus, il est possible de définir deux formulations différentes de cette approche à base de modèles : l'approche ***FDI*** *(Fault Detection and Isolation)* issue de la communauté Automatique, s'appuyant sur des modèles quantitatifs, et l'approche ***DX*** (du nom du *International workshop on principles of diagnosis*) de la communauté de l'Intelligence Artificielle utilisant des modèles qualitatifs.

1.3.1.1 Modèles Quantitatifs

Ces modèles sont construits à partir des lois fondamentales (physique, chimie,…) et décrits par des relations mathématiques sur les entrées-sorties du système. Diverses approches pour la détection de défaillances à partir des modèles mathématiques ont été développées depuis les années 70 [ISE97].

Ces méthodes dites "méthodes des résidus" comportent deux étapes : d'une part, la génération des ***résidus*** et, d'autre part, le choix d'une règle de décision pour le diagnostic. Les résidus représentent des changements ou divergences entre le comportement réel du processus et celui prévu par le modèle. La Figure 1.2 illustre le principe le plus général pour la génération des résidus.

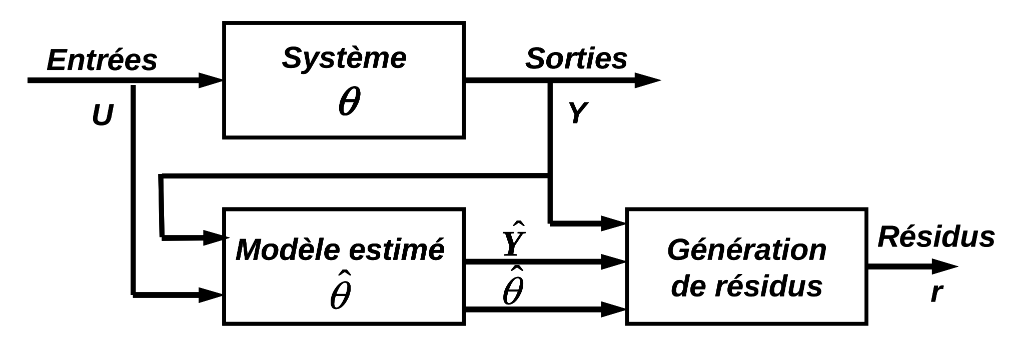


Figure 1.2. Principe de génération des résidus

L'objectif du résidu est d'être sensible aux défauts. Ainsi, normalement, en l'absence de défaillances, c'est-à-dire en fonctionnement normal, le résidu doit avoir une valeur nulle. Au contraire, en présence d'un défaut, le résidu aura une valeur non nulle.

Les techniques les plus utilisées pour la génération des résidus, à partir de modèles analytiques, sont listées ci-dessous :

- Équations de parité [GER98]
- Estimation d'état à partir d'observateurs ou filtres de Kalman [FRA89]
- Estimation paramétrique [ISE93]
- Analyse structurelle [CASS97]

Une fois les résidus générés, ils doivent être évalués pour déterminer la présence ou non d'une défaillance. Cette évaluation des résidus est établie principalement par l'utilisation de seuils fixes ou adaptatifs pour éviter les fausses alarmes. Néanmoins, la plupart du temps, les résidus sont corrélés entre eux. Pour gérer cette corrélation, la *maximum de vraisemblance généralisée* peut être utilisée. Il s'agit d'une technique qui, sous l'hypothèse que les variables ont une distribution connue, usuellement la distribution normale, permet d'estimer les paramètres d'un modèle (d'une équation ou d'un système, linéaire ou non linéaire) avec des restrictions sur les paramètres (coefficients, matrice de variances et covariances) ou non. Plus spécifiquement, la technique consiste à construire une fonction appelée fonction de vraisemblance (construite à partir de la fonction de densité) et à maximiser son logarithme par rapport aux paramètres inconnus.

Pour isoler la faute, l'approche *FDI* utilise une ***matrice de signatures de fautes (Σ)*** qui est obtenue à partir de l'ensemble des résidus. Cette matrice décrit les relations entre les résidus (lignes de Σ) et les fautes (colonnes de Σ). Un élément de Σ_{ij} aura une valeur de 1 si la faute de la colonne j a une influence sur le résidu de la ligne i, dans le contraire, la valeur sera 0. La comparaison d'une signature observée avec les différentes colonnes de la matrice permet de localiser la faute.

Le principal inconvénient des méthodes analytiques de détection de défaillances et diagnostic est la nécessité d'avoir des modèles mathématiques assez précis et complets, ce qui n'est pas toujours facile, voire impossible, pour des processus complexes tels que les processus chimiques. Ces modèles sont limités aux représentations linéaires ou à des modèles non-linéaires très spécifiques. Un autre inconvénient est la modélisation des perturbations qui peuvent engendrer des erreurs dans le modèle. En plus, si un type de faute n'a pas été modélisé de manière spécifique, il n'y a pas de garanties que les résidus soient capables de le détecter. Finalement, l'adaptabilité de ces approches aux changements du processus n'est pas considérée.

1.3.1.2 Modèles qualitatifs ou semi-qualitatifs

Dans certains cas, il devient très difficile de disposer des connaissances complètes pour faire un modèle analytique du processus à cause de sa complexité structurelle ou de son comportement. Une alternative pour traiter les connaissances incomplètes est de faire des abstractions selon le principe du ***raisonnement qualitatif*** [TRA97a]. Les modèles décrivent alors la *structure* du processus à diagnostiquer. Généralement, il s'agit des connexions entre les *composants.* L'approche *DX* exprime explicitement le lien entre un composant et les formules décrivant son comportement. Ces dernières années, l'utilisation de modèles qualitatifs pour la surveillance et le diagnostic de défaillances a pris une grande ampleur [TRA97b] [ESC01].

Parmi les méthodes qui utilisent les modèles physiques de type qualitatif pour la détection et le diagnostic de fautes, se trouve la prédiction du comportement à partir des équations différentielles qualitatives (***QDEs***). Ces QDEs sont des abstractions des équations différentielles ordinaires qui utilisent des intervalles de variables et des paramètres. L'algorithme ***QSIM*** [KUI86] représente un système comme un ensemble d'équations différentielles qualitatives (QDEs) couplées. Le comportement du système est alors représenté par une séquence d'états.

Selon l'approche *DX,* le diagnostic est basé sur la théorie du raisonnement logique [DAG01]. Pour le diagnostic, les termes ***suspect*** et ***candidat*** sont utilisés. Le premier fait référence à tout composant que le système de diagnostic a identifié comme responsable possible de l'apparition d'une divergence. Tandis que le deuxième fait référence au composant, ou ensemble de composants, dont le fonctionnement incorrect expliquerait toutes les divergences observées. Une caractéristique de cette technique est que le diagnostic se fait de façon itérative en trois phases :

- *La Détection de conflits* : recherche de l'ensemble de composants suspects en fonction des divergences observées.

- *Le Diagnostic* : recherche des candidats parmi les suspects.
- *La Discrimination des hypothèses :* raffinement de l'ensemble de candidats dans le cas où il y en aurait plusieurs.

1.3.1.3 Systèmes flous

Une autre alternative pour la modélisation des systèmes complexes est l'utilisation des concepts de la théorie des ensembles flous et de la logique floue [YAG94]. Un modèle flou est une représentation des caractéristiques d'un système à l'aide de règles floues qui décrivent sont comportement. Ceci permet une généralisation de l'information qui essaie d'imiter le raisonnement approximatif exécuté par l'homme en introduisant l'imprécision [ZAD73]. Les systèmes à base de règles floues sont des systèmes où les relations entre variables sont représentées au moyen de règles floues de la forme : ***SI prémisse ALORS conclusion***. Fondamentalement deux types de modèles sont distingués :

- *Modèles linguistiques (ML) flous* [MAM77] qui décrivent le système à partir des règles *SI*-ALORS où, les prémisses et les conclusions utilisent des variables linguistiques qui ont des ensembles flous comme valeurs. L'ensemble de règles prend la place de l'ensemble d'équations classique utilisé pour caractériser un système. Ce type de modèles est essentiellement une expression qualitative du comportement du système, où les règles sont obtenues la plupart du temps à partir de la connaissance des experts et des différents mécanismes de raisonnement. De façon générale une règle *SI*-ALORS linguistique s'écrit :

$$R_i : \quad SI \ x \ est \ A_i \ ALORS \ y \ est \ B_i \quad i = 1,2,...,K \tag{1}$$

  Où A_i et C_i sont des ensembles flous qui définissent le partitionnement des espaces d'entrée et de sortie.

- *Modèles flous type Takagi-Sugeno-Kang (TSK)* [TAK85] pour lesquels les prémisses des règles sont de variables linguistiques mais les conclusions sont de type numérique, généralement sous forme de constantes, de polynômes ou de fonctions (linéaires ou non-linéaires) qui dépendent des entrées :

$$R_i : \quad SI \ x \ est \ A_i \ ALORS \ y_i = f_i(x), \quad i = 1,2,...,K \tag{2}$$

  Les modèles flous *TSK* permettent une application relativement facile des techniques puissantes d'apprentissage pour leur identification à partir des données. Ce type de modèle est utilisé pour approximer des systèmes complexes non linéaires. En effet, le grand avantage du modèle *TSK* se situe dans sa puissance de représentation, particulièrement pour décrire des processus complexes. Il permet de décomposer un système complexe en sous-systèmes plus simples (parfois même linéaires). La construction d'un tel système flou est une chose délicate. Quand la connaissance d'un expert est disponible les règles doivent être exprimées de manière linguistique, ce qui amène ensuite à déterminer des fonctions d'appartenance pour les prémisses. En ce qui concerne les conclusions, les

fonctions f_i sont typiquement des instances d'une fonction paramétrique, dont la structure demeure la même dans toutes les règles, seuls les paramètres changent.

1.3.2 Méthodes à base de connaissances

Nous venons de voir que, dans le domaine de l'automatique et de la supervision des processus, la conception et l'utilisation des modèles mathématiques précis pour la détection et le diagnostic est bien connue. Mais, dans beaucoup de cas, la construction d'un tel modèle est très difficile, de par la nature complexe ou non-linéaire du processus lui-même, des paramètres variables dans le temps ou du manque de mesures disponibles. Dans la pratique, il est démontré que, dans ce cas, l'opérateur humain peut fournir une meilleure supervision en utilisant sa propre connaissance et son expérience pour assurer le bon fonctionnement du processus. De ce fait, les connaissances de l'expert sur la structure et le comportement du processus sont exploitées par diverses techniques.

1.3.2.1 AMDEC et Arbres de défaillances

Nous présentons ces méthodes, issues des études de risques et de sûreté de fonctionnement, qui peuvent être utilisées pour identifier les causes des défaillances de processus industriels. Il s'agit, en fait, de l'analyse fonctionnelle et matérielle du système.

- **AMDE** (Analyse des Modes de Défaillance, de leurs Effets)

La méthode de l'Analyse des Modes de Défaillance et de leurs Effets fût employée pour la première fois, à partir des années 1960, dans le domaine de l'aéronautique pour l'analyse de la sécurité des avions [REC66]. Cette méthode permet une analyse systématique et très complète, composant par composant, de tous les modes de défaillance possibles et précise leurs effets sur le système global [VIL88].

La démarche consiste d'abord à définir le système, ses fonctions et ses composants. Ensuite, l'ensemble des modes de défaillances des composants doit être établi. Pour chaque mode de défaillance, sont recherchées ensuite les causes possibles de son apparition. Finalement, une étude des effets sur le système et sur l'opérateur est faite pour chaque combinaison (cause, mode de défaillance). Les résultats sont présentés sous forme de tableau.

L'AMDE est parfois complétée par une analyse de criticité ; elle devient AMDEC (Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité). La criticité permet d'extraire les modes de défaillance les plus critiques. L'analyse de criticité est évaluée à partir du couple probabilité-gravité.

L'*Analyse de fonctionnement (HAZOP, Hazard and operability study)* est une autre extension de l'analyse des modes de défaillance (AMDE). La méthode HAZOP explore systématiquement l'aspect fonctionnel d'un système en identifiant, à la fois, les dangers et les dysfonctionnements d'une installation. Elle est mise en oeuvre à la fin de la phase de conception puisqu'elle s'appuie sur les schémas détaillés du système.

L'utilisation des tableaux d'AMDE(C) à des fins de diagnostic industriel conduit à utiliser une procédure déductive, c'est-à-dire à utiliser ces tableaux comme un outil d'identification des causes de défaillances à partir des effets observés [ZWI95]. La modélisation des relations cause à effet réalisée par l'AMDEC rend cette démarche très puissante pour la résolution des problèmes de diagnostic de défaillances de procédés industriels.

L'AMDE est donc très répandue dans des nombreux domaines industriels, l'aéronautique en particulier ; pourtant elle est lourde et insuffisante. En effet, il est nécessaire d'identifier *a priori* les défauts et/ou les dysfonctionnements pouvant apparaître, à un certain moment, dans le système supervisé; le recensement préalable des défaillances et leurs relations éventuelles ne peut jamais être exhaustif et requiert en général une longue expérience. En plus, toute modification ou évolution du système nécessite une réécriture du tableau. Enfin, cette méthode ne peut traiter les cas de défaillances multiples et intégrer l'aspect fonctionnel.

- **Arbres de Défaillances** (ou Arbres de Causes)

L'arbre de défaillance est l'un des outils majeurs d'analyse du risque technologique, développé au début des années 1960 aux Etats-Unis [VIL88]. C'est la méthode la plus couramment utilisée dans les analyses de fiabilité, de disponibilité ou de sécurité des systèmes.

Il s'agit d'une méthode optimisée qui permet la détermination des chemins critiques dans un système. Cette méthode a pour objet de déterminer les diverses combinaisons possibles d'événements qui entraînent la réalisation d'un événement indésirable unique. Elle permet donc d'identifier les points faibles de la conception.

C'est une méthode déductive dont la représentation graphique des combinaisons est réalisée par une structure arborescente (arbre), permettant un traitement à la fois qualitatif et quantitatif.

Cet arbre est établi sous forme d'un diagramme logique et comporte au sommet l'événement indésirable. Les causes immédiates qui produisent cet événement sont ensuite hiérarchisées à l'aide de symboles logiques "ET" et "OU". De cette façon, l'arbre est créé pas à pas pour atteindre, à la base, un ensemble d'événements considérés comme élémentaires.

Une extension des arbres de défaillances est la méthode du diagramme de causes-conséquences [VIL88]. Cette méthode combine les principes utilisés par l'analyse déductive de l'arbre des défaillances et l'analyse inductive de l'arbre des conséquences.

Le principal inconvénient des arbres de défaillances est que le développement est sensible aux erreurs à différentes étapes. En effet, l'arbre construit est seulement aussi bon que le modèle mental de son créateur. Pour exécuter un diagnostic correct à partir des arbres de défaillances, ceux-ci doivent largement représenter toutes les relations causales du processus, c'est-à-dire qu'ils doivent être capables d'expliquer tous les scénarios de défauts possibles. De plus, l'emploi de cette méthode se révèle difficile pour les systèmes fortement dépendants du temps. Enfin, il n'y a pas de méthode formelle pour vérifier l'exactitude de l'arbre développé.

1.3.2.2 Systèmes experts

Pour les systèmes experts, ou systèmes *à base de connaissances,* le diagnostic industriel est devenu un des domaines privilégiés d'application. Les systèmes experts sont des outils de l'intelligence artificielle, utilisés lorsque aucune méthode algorithmique exacte n'est disponible ou praticable. De façon générale, nous pouvons dire qu'un système expert sert à codifier la connaissance humaine en termes d'expérience, raisonnement approximatif, analogie, raisonnement par défaut, apprentissage, etc. De ce fait, la propriété principale de ces systèmes est de pouvoir représenter et restituer les connaissances acquises par les spécialistes d'un domaine technique précis. Les connaissances utilisées, dans la plupart des cas, pour le développement d'un système expert d'aide au diagnostic, reposent sur l'apprentissage des relations entre les causes et les effets observés pour chaque défaillance [AGUI99]. Néanmoins, il est possible aussi d'utiliser les modèles fonctionnels décrivant les comportements des composantes de systèmes complexes [CHA93]. La définition que nous retenons pour les systèmes experts est celle donnée par [ZWI95]:

> « *Un système expert est un système informatique destiné à résoudre un problème précis à partir d'une analyse et d'une représentation des connaissances et du raisonnement d'un (ou plusieurs) spécialiste(s) de ce problème* »

Un système expert est composé de deux parties indépendantes (voir Figure 1.3) :

- une **base de connaissances** elle-même composée d'une **base de règles** qui modélise la connaissance du domaine considéré et d'une **base de faits** qui contient les informations concernant le cas traité,
- un **moteur d'inférences** capable de raisonner à partir des informations contenues dans la base de connaissances, de faire des déductions, etc. Au fur et à mesure que les règles sont appliquées des nouveaux faits se *déduisent* et se rajoutent à la base de faits.

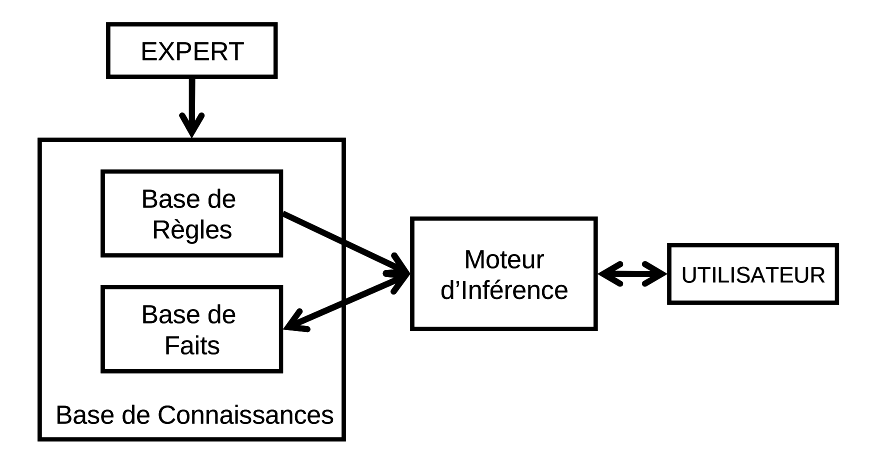


Figure 1.3. Architecture d'un système expert

Les principaux avantages des systèmes experts pour le diagnostic sont leur capacité à raisonner sous incertitude et leur capacité à apporter des explications des solutions fournies. La difficulté spécifique de leur mise en œuvre est la formalisation de la démarche cognitive qui a pour objectif, à partir d'une situation donnée, de définir et de décrire le raisonnement associé.

1.3.3 Méthodes à partir des données historiques

Ces méthodes ont l'avantage de ne pas nécessiter la connaissance d'un modèle mathématique ou structurel du procédé, contrairement aux méthodes à base de modèles. Seulement, la disponibilité de grandes quantités de données historiques du processus est nécessaire. Il y a diverses façons de traiter ces données et de les présenter en tant que connaissance a priori pour le diagnostic. Ceci est connu comme l'***extraction des caractéristiques****.* Parmi les méthodes d'extraction des caractéristiques, se trouvent les méthodes de représentation et analyse qualitative des tendances, l'Analyse en Composantes Principales (**ACP**)/Moindres Carrés Partiels (**MCP**) et les méthodes de classification et de reconnaissance de formes (**RdF**), parmi lesquelles nous pouvons citer, d'une part, les classificateurs statistiques, d'autre part, les réseaux de neurones artificiels [VEN03b].

1.3.3.1 Analyse et représentation qualitative des tendances

Un traitement général des signaux a pour but l'analyse et la représentation qualitative des tendances du processus, proposée par Cheung et Stephanopoulos [CHE90] avec le concept de ***tendance***, série *d'**épisodes*** avec une représentation triangulaire. Dans ce formalisme, chaque épisode est représenté par sa pente initiale, sa pente finale (à chaque point critique) et un segment de droite reliant les deux points critiques. Janusz et Venkatasubramanian [JAN91] ont proposé une description qualitative des signaux (*TDL – Trend Description Language*) utilisant *primitives, épisodes, tendances* et *profils*. Les *primitives* sont basées sur le signe des première et deuxième dérivées (positif, zéro et négatif). Ceci donne un ensemble de 9 primitives. Un épisode est un ensemble de primitives du même type. Le nombre de fois consécutives qu'une primitive est présente définie la durée de l'épisode. La *tendance* d'un *signal* consiste en une série d'*épisodes*, et un *profil* s'obtient en ajoutant des informations quantitatives aux tendances. Par la suite, plusieurs travaux ont amélioré le langage de description des tendances (TDL) au niveau du bruit [VED97], de sa représentation et de son interprétation [AYR95][MEL01] pour l'utiliser, ensuite, en combinaison avec d'autres techniques, surtout des techniques de reconnaissance de formes [BAK92].

Afin d'être plus générale, l'approche décrite par Meléndez et Colomer [MEL01] a élargit les formalismes précédents vers des contextes qualitatifs et quantitatifs. Cette approche permet, en effet, de construire des épisodes à partir de toute caractéristique extraite des variables du processus. Selon ce formalisme, une nouvelle représentation permet de décrire les tendances d’un signal selon sa deuxième dérivée, laquelle peut être calculée par le biais d'un différentiateur (« band-limited » : FIR) [COL01] afin d'éviter l'amplification de bruit. La première dérivée qualifiée au début et à la fin de chaque épisode est employée pour obtenir une représentation plus significative de la tendance du signal. Ainsi, un ensemble de 13 types d'épisodes est obtenu (voir Figure 1.4). Un avantage important de cet ensemble d'épisodes pour la surveillance est que les discontinuités et les périodes de stabilité (habituelles dans des situations de défaillance et dans des situations normales respectivement) sont explicitement représentées en utilisant 5 types d'épisodes ($\alpha, \beta, \gamma, \lambda, \mu$). Cette approche a été utilisée,

en combinaison avec d'autres méthodes, pour le diagnostic d'unités industrielles [COL02].

A	B	X	Δ	E	ϕ	γ	η	ι	φ	κ	λ	μ
A	B	C	D	E	F	G	H	I	J	K	L	M

Figure 1.4. Types d'épisodes à partir de la deuxième dérivée

L'abstraction qualitative permet une représentation compacte de la tendance en ne prenant en compte que les événements significatifs. Pour des tâches telles que le diagnostic, la représentation qualitative des tendances fournit souvent des informations valables qui facilitent le raisonnement sur le comportement du processus. Dans la plupart des cas, les dysfonctionnements du procédé produisent une tendance distinctive dans les signaux issus des capteurs. Ces tendances peuvent être employées pour l'identification des situations anormales dans le processus. Ainsi, une analyse appropriée des tendances du processus peut permettre de détecter plus tôt une défaillance.

1.3.3.2 Analyse en Composantes Principales (ACP)

L'analyse en composantes principales (**ACP**) est une technique statistique multi-variable. Les techniques statistiques multi-variables sont des outils puissants, capables de comprimer des données et de réduire leur dimensionnalité de sorte que l'information essentielle soit maintenue et plus facile à analyser que dans l'ensemble original des données. Ces techniques peuvent également manipuler le bruit et la corrélation pour extraire l'information efficacement. La fonction principale de ce type de techniques est, par le biais d'une procédure mathématique, de transformer un certain nombre de variables corrélées en un ensemble plus petit de variables non corrélées.

L'ACP est essentiellement basée sur une décomposition orthogonale de la matrice de covariance des variables du processus le long des directions qui expliquent la variation maximale des données, c'est-à-dire que cette méthode recherche une projection des observations sur des axes orthogonaux. De ce fait, le premier axe contient la plus grande variation. Le deuxième axe contiendra la seconde plus grande variation orthogonale au premier.

Le but principal de l'ACP est donc de trouver un ensemble de facteurs (composantes) qui ait une dimension inférieure à celle de l'ensemble original de données et qui puisse décrire correctement les tendances principales. L'ACP est une procédure qui ne prend en compte que les variables du processus. Parfois, un ensemble additionnel de données est disponible, e.g. variables de qualité du produit. Il est souhaitable d'inclure toutes les données disponibles pour la surveillance du procédé et d'utiliser de cette façon les variables du processus pour prédire et détecter des changements dans les variables de qualité du produit. Pour cela, la méthode des *Moindres Carrés Partiels (MCP)* peut être utilisée [VEN03b]. Cette méthode modélise le rapport entre deux blocs de données tout en les comprimant simultanément. Elle est employée pour extraire les variables latentes qui expliquent la variation des données du processus.

Divers travaux menés par MacGregor *et al.* [MCG95][NOM94] ont montré l'application des méthodes statistiques multi-variables telles que les ACP et MCP, dans l'analyse et la commande de processus, la détection et le diagnostic de défaillances, aussi bien dans le cas de procédés continus que dans les cas des procédés "*batch*" (continus par lots).

Une limitation importante de la surveillance basée sur l'ACP est que la représentation obtenue est invariante dans le temps, tandis que la plupart des processus réels évoluent au cours du temps. Par conséquent, la représentation issue de l'ACP nécessite également d'être mise à jour périodiquement. Un autre inconvénient est qu'elle ne possède pas des propriétés de *signature* pour le diagnostic, ce qui rend l'isolation des défaillances difficile.

1.3.3.3 Classification et méthodes de reconnaissance de formes

Ici, le diagnostic de défaillances est essentiellement vu comme un problème de classification. Le but principal est de construire un bloc de correspondance tel qu'à partir d'un ensemble d'informations décrivant la situation courante du processus, il est possible d'obtenir les causes probables des situations anormales. Or, quand le diagnostic est basé sur des observations multiples, ces observations sont regroupées pour former des classes qui définissent une situation ou un mode de fonctionnement du processus, auxquelles une nouvelle observation sera comparée pour être identifiée. En d'autres termes, le diagnostic a pour mission d'identifier le mode de fonctionnement d'un système à partir d'observations sur celui-ci.

La *classification* peut être définie comme le groupement des *observations* (ou objets ou individus) qui ont des caractéristiques similaires. Chaque observation est définie par un vecteur noté $\boldsymbol{X}^{\boldsymbol{i}} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{\boldsymbol{D}}]^{\boldsymbol{T}}$ dans un espace à *D* dimensions, où *D* correspond au nombre d'attributs ou caractéristiques. Cet espace est connu sous le nom d'***espace de représentation*** ou de ***description***. Le regroupement des observations selon leurs caractéristiques permet la définition des ***classes*** décrivant des régions particulières de l'espace de représentation. L'ensemble des classes $\boldsymbol{C} = \{\boldsymbol{C}^{\boldsymbol{k}}, \boldsymbol{k} = 1 \ldots \boldsymbol{K}\}$ définit l'***espace de décision***. Le regroupement des observations pour construire l'espace de décision est fait à partir d'une phase d'apprentissage en utilisant une mesure de ***similitude*** par le biais des ***méthodes de classification.***

L'objectif de la reconnaissance de formes, illustré sur la Figure 1.5, est alors de savoir associer toute nouvelle observation $\boldsymbol{X}^{\boldsymbol{i}}$ à une classe de l'espace de décision. L'affectation d'une observation $\boldsymbol{X}^{\boldsymbol{i}}$ à l'une des ***K*** classes notées $\boldsymbol{C}^1, \ldots, \boldsymbol{C}^{\boldsymbol{K}}$ indique une opération de *classement* ou de *discrimination.*

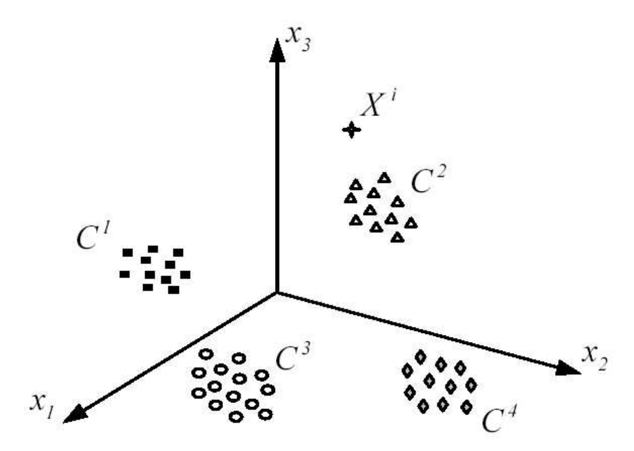


Figure 1.5. Objectif en Reconnaissance des Formes : associer une nouvelle observation X^i à une des classes.

Si nous nous plaçons dans le contexte du diagnostic, les paramètres caractérisant le vecteur d'observation, et par conséquent l'espace de représentation, seront les mesures disponibles issues des capteurs et actionneurs du processus ou bien des informations extraites de ceux-ci. Seule la connaissance a priori sur le système permet de choisir quelles seront les caractéristiques à retenir pour le diagnostic. Les classes peuvent être assimilées aux différents modes de fonctionnement du processus, définissant alors l'espace de décision. Donc, l'objectif en termes de diagnostic est de définir à quel mode de fonctionnement correspond une nouvelle observation.

Dans la plupart des cas, la conception d'un système de diagnostic à partir de la reconnaissance de formes se déroule en trois étapes [DUB90] :

- Une **phase d'analyse** où il s'agit d'étudier les informations fournies par les différents capteurs, les historiques ainsi que l'expérience des opérateurs, afin de définir l'espace de représentation du système ainsi que les classes qui représenteront les différents états de fonctionnement. Il s'agit donc de la phase de classification : un ensemble de ***N*** observations $X^1, X^2, \dots X^N$ qui seront regroupées en ***K*** classes. Bien que cela dépende de la méthode de classification utilisée pour l'apprentissage, cette phase est généralement lourde en terme de calcul.
- Une **phase de choix** du système de détection qui consiste à construire une règle de décision qui établira des frontières entre les différentes classes. La règle de décision permettra d'affecter ou non une nouvelle observation à l'une des classes connues.
- Une **phase d'exploitation** qui consiste à implémenter le système afin de proposer une décision pour toute nouvelle observation effectuée. La bonne exploitation du système dépend de la pertinence de l'espace de représentation et des performances de la règle de décision.

Un des points les plus importants dans les méthodes de reconnaissance de formes, est le bon choix de l'espace de représentation, car de celui-ci dépend la qualité du résultat. Malheureusement, il n'existe pas de règle générale pour choisir l'espace de représentation, ce choix s'appuie sur les connaissances a priori existantes sur le

problème à résoudre; il n'y a pas de démarche algorithmique qui permette de le formaliser.

Un des inconvénients d'utiliser les méthodes de reconnaissance des formes pour le diagnostic est que, en général, le nombre de classes est supposé connu au départ, i.e. la connaissance est supposée exhaustive. Toute observation doit appartenir à une des classes définies. Ceci est loin d'être le cas dans le contexte du diagnostic. Par ailleurs, il reste souvent des modes de fonctionnement dont nous ignorons les caractéristiques. Dans ce cas, il est nécessaire d'utiliser des techniques qui permettent de prendre en compte la détection de nouvelles classes, d'apprendre leurs caractéristiques et de les inclure dans le système de détection; ce qui implique qu'un système de diagnostic basé sur des méthodes de reconnaissance de formes doit être un système évolutif, capable de prendre en compte tout nouveau phénomène normal ou anormal présent dans le processus surveillé.

De nombreux travaux ont déjà permis de montrer l'intérêt des méthodes de classification pour le diagnostic, notamment, dans le cadre de systèmes complexes [RIB02]. Les méthodes choisies vont des méthodes les plus classiques, comme les classificateurs statistiques (le classificateur Bayesien, les *K* plus proches voisins) [MAR03][CAS03], jusqu'aux réseaux de neurones artificiels [VEN90][LUR01]. Nous reviendrons plus en détails sur ces différentes méthodes. (§ 2.3)

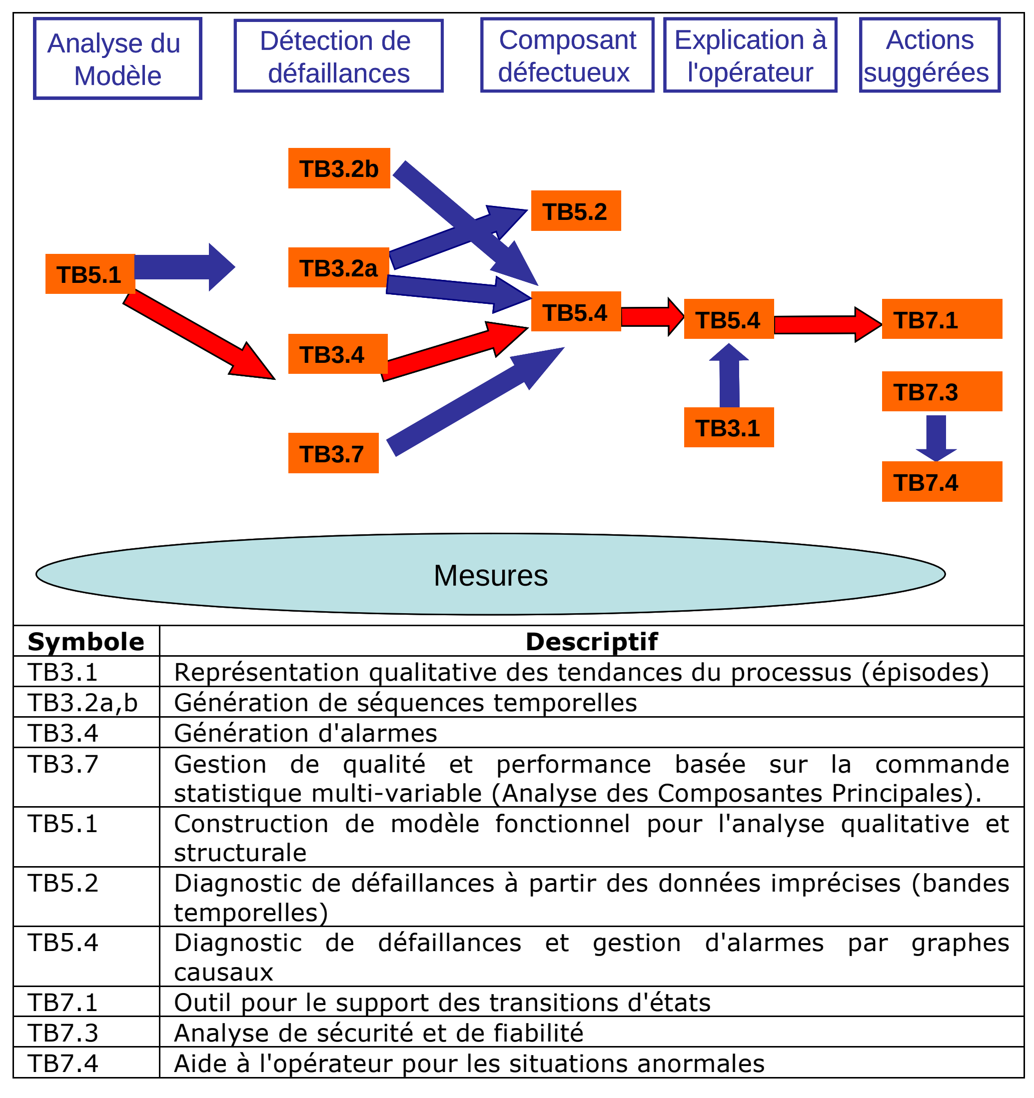
1.4 Tentative d'intégration des méthodes et outils

Il devient évident que toutes ces différentes méthodes de détection et diagnostic, évoquées précédemment, ont leurs points forts et leurs faiblesses. Ainsi, il est possible que certaines méthodes puissent se complémenter laissant espérer de meilleurs systèmes de diagnostic. L'intégration de diverses méthodes complémentaires est une façon de développer des systèmes "hybrides" permettant de surmonter les limitations des stratégies individuelles. Par exemple, les tendances qualitatives (épisodes) peuvent être utilisées en complément d'une base de connaissance à base de règles floues (*SI-ALORS)* pour réaliser l'affectation des tendances à différentes situations ou états et l'identification d'une défaillance dans le processus [DAS03]. Isermann [ISE97] a proposé, dans sens, l'utilisation de modèles analytiques et heuristiques pour la création de symptômes et, à partir de ces symptômes, de faire du diagnostic, c'est-à-dire d'affecter les symptômes aux défaillances, en utilisant des techniques de classification ainsi que des méthodes d'inférence.

En ce qui concerne le cas des procédés chimiques complexes, Ruiz *et al.* [RUI01] ont développé un système d'aide pour le diagnostic de défaillances en combinant une méthode de reconnaissance de formes (les réseaux de neurones (RN)), avec un système à base de connaissances (SBC) construit à partir de règles floues. Le RN utilise les données historiques ainsi que les données issues d'un simulateur construit avec un modèle du procédé, pour apprendre les situations normales et anormales. Le SBC, lui, est construit avec un modèle HAZOP du processus et les connaissances de l'expert. Les

sorties du RN qui donnent des informations sur les fautes possibles, sont utilisées comme des entrées pour le SBC.

Aujourd'hui, les statistiques industrielles estiment que le coût économique des situations anormales est autour des 20 milliards d'euros et ce, seulement, dans l'industrie pétrochimique [NIM95]. De ce fait, plusieurs initiatives de collaboration entre les industriels et les communautés de recherche scientifique visent aux développements d'outils pour améliorer la supervision des unités complexes. Une de ces initiatives est le ***Projet CHEM*** *(Advance Decision Support System for chemical/petrochemical manufacturing processes)* [CHEM]. Le but principal de ce projet est de regrouper, à partir d'une intégration modulaire d'outils logiciels, des techniques de supervision et d'aide à la décision, notamment de détection, de diagnostic et de reconfiguration, développées par différentes équipes de recherche européennes. Cette intégration modulaire permet le choix et la combinaison des techniques les plus appropriées, selon les connaissances disponibles du processus à superviser, afin de donner à l'opérateur des informations représentatives pour la prise de décisions. L'annexe A de cette thèse est dédiée à la présentation plus détaillée du projet CHEM.



Symbole	Descriptif
TB3.1	Représentation qualitative des tendances du processus (épisodes)
TB3.2a,b	Génération de séquences temporelles
TB3.4	Génération d'alarmes
TB3.7	Gestion de qualité et performance basée sur la commande statistique multi-variable (Analyse des Composantes Principales).
TB5.1	Construction de modèle fonctionnel pour l'analyse qualitative et structurale
TB5.2	Diagnostic de défaillances à partir des données imprécises (bandes temporelles)
TB5.4	Diagnostic de défaillances et gestion d'alarmes par graphes causaux
TB7.1	Outil pour le support des transitions d'états
TB7.3	Analyse de sécurité et de fiabilité
TB7.4	Aide à l'opérateur pour les situations anormales

Figure 1.6. Structure d'intégration pour le projet CHEM

La Figure 1.6 présente la structure générale d'intégration et de complémentarité des différentes techniques telle que réalisée dans le projet CHEM. Dans ce schéma type, les résultats des outils de détection sont utilisés comme des entrées pour les outils de diagnostic dont les sorties sont ensuite transmises aux outils de support à l'opérateur.

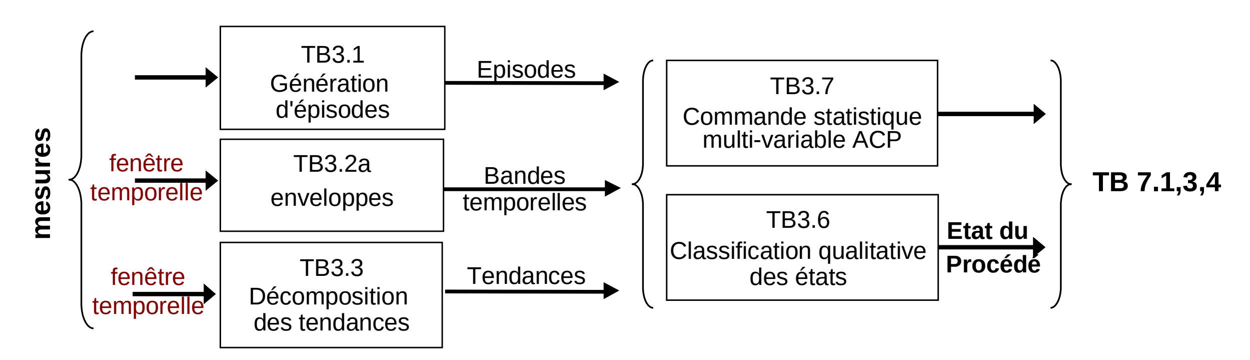


Figure 1.7. Interactions des outils à base de données historiques

Une autre approche d'intégration des méthodes à base de données historiques est présentée dans la Figure 1.7

1.5 Conclusion

L'objectif de ce premier chapitre était, d'une part, d'identifier une problématique générale au niveau du fonctionnement des applications industrielles, d'autre part, de faire ressortir la nécessité de mettre en place des systèmes de supervision pour réduire les coûts économiques et répondre aux contraintes environnementales et de sécurité. Nous avons proposé un rapide état de l'art des différentes méthodes de diagnostic de défaillances pour des applications industrielles. Trois catégories de méthodes ont été présentées : les méthodes à base de modèles, les méthodes à base de connaissances et les méthodes à partir des données historiques. Le choix d'une de ces méthodes dépend essentiellement des connaissances disponibles sur le procédé, sans oublier les considérations techniques et économiques. Néanmoins, nous avons constaté que ces méthodes ont des limitations et qu'un cadre de travail pour la résolution des problèmes de façon collective, utilisant des raisonnements différents et parallèles, s'avère être une alternative attractive pour relever les défis du diagnostic d'unités industrielles complexes.

Dans le cadre de cette problématique, notre travail porte sur le développement d'un outil d'aide pour la caractérisation et l'identification du comportement du procédé. Ceci est réalisé au moyen des méthodes de classification, de façon à exploiter les vastes quantités de données issues des capteurs et à prendre en compte les connaissances de l'opérateur "expert". Nous nous sommes intéressée aux méthodes de classification et reconnaissance de formes en considérant le fait que, même s'il n'y a pas une connaissance exhaustive de tous les modes de fonctionnement normaux ou

anormaux du processus, l'outil de supervision doit être en mesure de détecter des nouveaux phénomènes et les prendre en compte de façon à les caractériser et à les intégrer. Autrement dit, l'outil recherché doit être **adaptatif**.

Il apparaît donc, clairement, que le fait d'être capable d'identifier la situation dans laquelle se trouve le procédé à chaque instant est indispensable pour apporter des informations supplémentaires sur les causes d'une défaillance et pour aider dans la prise de décision des actions à mener pour la reconfiguration du système. Mais il est également prépondérant que l'opérateur puisse suivre l'enchaînement des situations et les changements d'état à la suite d'un événement

Les outils de supervision doivent donc, non seulement **aider** à la reconnaissance des comportements ("*situation assessment*"), mais aussi **fournir** à l'opérateur les moyens d'avoir un suivi de ces comportements, une sorte de "modèle de comportement".

Ainsi, dans le chapitre suivant, nous présentons une stratégie pour la construction d'un modèle de comportement d'un processus. Ce modèle est obtenu à partir des données, en utilisant des méthodes de classification. Dans un premier temps, les différentes étapes à prendre compte pour la caractérisation et l'identification des situations du processus sont présentées. Par la suite, différentes techniques de classification sont exposées afin de choisir celle qui semble la plus appropriée pour la mise en place de la stratégie.

Chapitre 2

2. UNE STRATEGIE POUR LA SURVEILLANCE A BASE DE METHODES DE CLASSIFICATION

2.1 Introduction

Comme nous l'avons dit précédemment, un système de surveillance doit être capable de rendre compte de l'état d'un procédé à tout moment. Un système de surveillance est une source de gains potentiels pour une entreprise : des gains financiers, en diminuant les pertes de production, les rebus et le coût d'entretien des équipements, entre autres; des gains matériels en minimisant les détériorations prématurées des machines, en évitant l'apparition de fautes plus graves; et surtout des gains "humains", en réduisant les risques auxquels sont exposés les opérateurs.

Notre objectif est de développer une stratégie permettant, à partir de données historiques et de données récupérées en ligne lors de l'exploitation du procédé, de construire un modèle du comportement du processus et d'identifier des situations anormales issues des dysfonctionnements, pour aider l'opérateur humain dans sa prise de décisions. Plus précisément, le but est d'identifier, au plus tôt, les défaillances du processus surveillé, en diminuant le nombre de fausses alarmes.

2.2 Une stratégie pour la détection et le diagnostic

L'objectif de cette démarche est double :

1. Il s'agit de suivre l'évolution temporelle des modes de fonctionnement identifiés du processus, c'est-à-dire de fournir à l'opérateur des informations appropriées chaque fois qu'une nouvelle situation se présente ainsi que des caractéristiques sur cette situation. Une situation peut se définir comme le résultat d'une évolution du processus et correspond à un état ou mode de fonctionnement du processus étudié. En prenant en compte le fait qu'une défaillance peut se produire à tout moment sur le processus, ces états peuvent être soit des états de fonctionnement normaux soit des états anormaux. Parce que dans la plupart des processus complexes il n'existe pas une connaissance exhaustive des situations de dysfonctionnement, nous considérons que toute situation inattendue est une situation anormale.

2. Le suivi des évolutions permet, alors, de réaliser la ***détection des dysfonctionnements*** en s'appuyant sur le principe de déviation du comportement observé par rapport au comportement prévu et non sur l'énumération a priori des situations de défaillances. La détection de ces états anormaux n'est pas suffisante et il est nécessaire de les caractériser de façon à identifier les causes, ce qui correspond à faire du ***diagnostic de défaillances***.

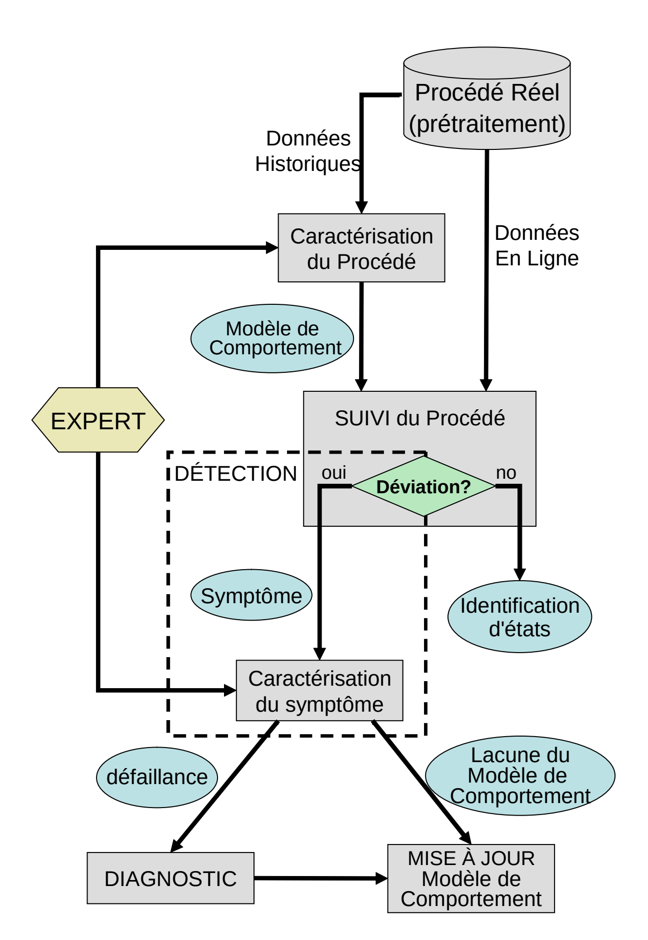


Figure 2.1. Stratégie pour l'identification des situations et la détection

La Figure 2.1 illustre un schéma général de la stratégie que nous proposons. Elle est basée sur l'analyse d'informations disponibles issues des capteurs du processus, ce qui nous a amené à nous pencher sur les méthodes à base de données historiques. La stratégie consiste en deux étapes différentes mais non-indépendantes. Une étape hors ligne dans laquelle des données historiques sont analysées et traitées pour caractériser le comportement connu du système, une deuxième étape dans laquelle le comportement du processus obtenu précédemment et les données en ligne sont utilisées pour déterminer l'état courant attendu du processus.

Dans la suite, nous donnons une description détaillée de chaque étape.

2.2.1 Caractérisation du procédé : Construction d'un modèle de référence (comportement)

Dans le cadre des processus complexes, nous avons déjà mentionné que la génération d'un modèle analytique approprié est un véritable défi. Une solution est d'employer les données historiques du procédé, souvent disponibles. Néanmoins, cette solution implique d'autres difficultés à gérer. En effet, la plupart du temps, les données sont hétérogènes et en grand nombre car elles résultent d'enregistrements réguliers. L'analyse de ces données devient une lourde tâche pour l'opérateur expert. En plus, comme toutes les décisions à prendre sur le processus dépendent de cette analyse, il est alors nécessaire d'aider l'expert pendant cette étape d'analyse. Comme nous l'avons précisé précédemment, au moyen des techniques d'apprentissage (*Machine Learning*) et de reconnaissance de formes, il est possible d'extraire des informations significatives sur le processus à partir des données historiques.

*L'**exploration des données** (Data Mining)*, ou extraction des connaissances à partir des données, est une façon d'aider l'expert. A partir des informations pertinentes issues de l'exploration des données historiques, un ***modèle de référence*** du comportement du procédé peut être obtenu. Ce modèle de comportement est élaboré hors ligne. Bien sûr, si les données historiques sont fiables, le modèle obtenu ne comportera pas d'erreurs de modélisation. Cependant, il ne sera pas exhaustif car les données historiques ne peuvent pas couvrir "la vie" entière du processus. Ce manque de complétude implique, comment nous l'avons déjà évoqué, qu'une situation anormale (non attendue) peut correspondre soit à une réelle situation de défaillance au niveau du système physique surveillé, soit à une situation de fonctionnement normal non présente dans le modèle de référence.

Dans certains cas, le modèle de référence peut inclure des situations identifiées avec assez de précision qui correspondent à des états critiques du processus, en terme de sûreté des équipements et/ou des opérateurs.

Pour l'élaboration du modèle de référence (comportement) à partir des données, une des étapes les plus importantes est la détermination de l'espace de représentation car du bon choix des données à utiliser, dépend la définition d'un ensemble de caractéristiques représentatives des états de fonctionnement du procédé. En outre, il

faut sélectionner une technique d'apprentissage assez flexible et adaptée aux données, capable de caractériser les différentes situations et d'en permettre une interprétation par l'expert. Nous reviendrons plus en détail sur les différents éléments de paramétrage à prendre en compte (voir §3.2) pour identifier les états fonctionnels en vue de la surveillance du procédé.

2.2.2 Surveillance et suivi du procédé

Une fois que la caractérisation du comportement du processus a été réalisée, une identification ou reconnaissance en ligne des états prévus (ou attendus) peut être faite. Ceci correspond à une étape de ***suivi*** du procédé. Pendant la reconnaissance, les données qui arrivent sont analysées pour décider si la situation correspond à un état de fonctionnement normal. L'identification de situations est alors basée sur la comparaison entre les différents états possibles du processus enregistrés dans le modèle de référence et les situations observées, décrites par les données en ligne.

Toute ***déviation*** du comportement prévu mène à la génération d'un ***symptôme***. Comme nous l'avons déjà évoqué, cette procédure de **détection** basée sur un tel principe de divergence (écart) laisse prendre en compte des situations anormales dues à de vraies défaillances sur les capteurs et/ou actionneurs du processus mais, également, des situations inattendues qui correspondent à un fonctionnement normal du processus et qui n'ont pas été considérées lors de l'élaboration du modèle de comportement.

À ce stade, la connaissance du symptôme n'est pas suffisante pour décrire la nature de la déviation observée. Une caractérisation du symptôme est alors nécessaire.

2.2.3 Détection des déviations : caractérisation des symptômes

La caractérisation des symptômes détectés issus des déviations du comportement attendu est effectuée hors ligne. Le but est de raffiner la détection de la situation anormale faisant une distinction entre une situation de défaillance et une situation normale mais non prévue. Pour ceci, il est d'abord nécessaire d'isoler les données associées à la déviation observée. Puis, plusieurs analyses sur ces données doivent être fournies à l'expert de telle manière qu'il puisse interpréter la signification du symptôme. Cette étape de **caractérisation**, cruciale dans la prise de décision des actions futures, demande un aménagement du dialogue avec l'expert.

Selon notre schéma de supervision de la Figure 1.1, un symptôme correspondant à une défaillance du processus doit activer une procédure de diagnostic. Le but du diagnostic, comme nous l'avons mentionné précédemment (voir § 1.2), est de trouver l'origine de la défaillance, c'est-à-dire trouver les causes de la situation de défaillance. Pour effectuer ce diagnostic d'autres informations ou connaissances du

processus peuvent être utilisées, tels que les HAZOP [VIL88], les graphes fonctionnels et/ou de composantes [TOG00].

Quel que soit le type de déviation observée, une défaillance ou une situation normale non prévue, elle doit être ajoutée dans le modèle de référence de manière à effectuer une mise à jour de celui-ci. De cette façon, un modèle de comportement du processus est construit itérativement.

2.3 Choix d'une méthode de classification pour l'implémentation de la stratégie

Si nous considérons les recherches menées par la communauté du ***Data Mining*** ainsi que la terminologie associée [FAY96], l'étape de caractérisation des modes de fonctionnement du processus, dont l'idée est déterminer les différentes régions de l'espace de représentation de façon à établir un modèle, correspond à la phase dite d'***apprentissage.*** La façon de réaliser la partition de l'espace, pour le regroupement des données, dépend du type de méthode utilisé pour l'apprentissage. Ainsi, la partition peut se faire de deux façons, en utilisant soit un apprentissage supervisé (dirigé), soit un apprentissage non-supervisé. Nous détaillerons ces aspects ultérieurement (§ 3.2.3). D'une manière analogue, l'étape du suivi et d'identification des situations attendues est équivalente à une phase de ***reconnaissance.*** Finalement, la caractérisation des symptômes peut être associée à une nouvelle procédure de regroupement des données avec des caractéristiques similaires (*"clustering"*) que nous avons appelée ***apprentissage supervisé actif*** car il s'agit de réaliser une nouvelle partition à partir des observations associées à la déviation, prenant en compte les classes établies auparavant, de façon à compléter et à faire évoluer le modèle de référence obtenu.

Il existe plusieurs techniques issues du *Data Mining* qui permettent la construction d'un modèle de comportement du processus par le biais d'une étape d'apprentissage.

En général, les méthodes de classification sont basées sur l'optimisation d'une *fonction objectif*, laquelle dépend du type de similitude défini dans l'ensemble des données (ou éléments) à classer. De ce fait, il existe des techniques qui minimisent les distances entre objets du même groupe et d'autres qui maximisent l'appartenance des objets à une classe.

Nous donnons, maintenant, une brève présentation des principales approches que nous considérons appropriées pour la mise en place de la stratégie proposée.

2.3.1 Méthodes de coalescence ("clustering")

Les techniques de coalescence (regroupement ou en anglais "*clustering*") essayent de trouver une partition de l'espace en regroupant les éléments, de nature quantitative $x \in \Re^D$, d'un ensemble $X = \{x_1, \ldots, x_N\}$ en un nombre ***K*** de groupes (ou classes), selon leurs ressemblances ou bien leur proximité [KAU90] [JAI99].

Typiquement, le concept de ***similitude*** est défini à partir de la **distance** entre un vecteur de données et le **prototype** (*centre*) du groupe (classe). Ces groupes doivent satisfaire deux critères :

- Chaque groupe ou classe est homogène : les objets appartenant au même groupe ou classe doivent être similaires entre eux (le plus proche possible).

- Chaque groupe ou classe doit être distinct des autres : les objets qui appartiennent au même groupe doivent être différents des objets appartenant à d'autres groupes (le plus éloigné possible des autres groupes).

Les caractéristiques des prototypes, qui ne sont pas habituellement connues à l'avance, doivent être choisies aléatoirement et sont mises à jour en même temps qu'est faite la partition de l'espace.

Les méthodes de regroupement sont basées sur l'optimisation itérative d'une fonction *objectif* (e.g. variabilité dans les classes), permettant d'obtenir des classes homogènes et les plus distinctes possibles.

2.3.1.1 K-Moyennes

L'algorithme populaire et le plus connu des méthodes de regroupement est celui des ***K-moyennes*** *(K-means)* [MCQ67]. Cet algorithme réalise une partition stricte ("dure"), c'est-à-dire que chaque objet n'est assigné qu'à une seule classe. Il s'agit d'une procédure simple et itérative dont l'idée générale est de classer un ensemble $X = \{x_1, ..., x_N\}$ d'éléments dans un nombre *K* de groupes ("clusters") fixés à l'avance.

Chaque élément est un vecteur $x \in \Re^D$. La partition est faite de telle façon que les éléments à l'intérieur d'un groupe sont les plus semblables possible, et les plus distincts des éléments appartenant à d'autres groupes. Le regroupement est réalisé en deux étapes : d'abord, il faut définir les *K* centres ou prototypes de chaque groupe; ensuite, chaque élément est associé au groupe dont il est le plus proche du centre. Ainsi, chaque groupe est caractérisé par son centre v_k et par les éléments qui lui ont été associés. Le centre du groupe est le point dans l'espace *D-dimensionnel* qui minimise la somme de la distance de tous les éléments qui font partie du groupe. Ici, *D* correspond au nombre de caractéristiques de l'espace de représentation.

L'algorithme a comme but la minimisation de la *fonction objectif :*

$$J = \sum_{k=1}^{K} \sum_{n=1}^{N} D_{knA}^2 \qquad (3)$$

Où D_{knA}^2 définit la mesure de distance entre l'objet x_n et le prototype v_k au sens d'une métrique définie par la matrice *A* :

$$D_{knA}^2 = \left\| x_n - v_k \right\|_A^2 = \left(x_n - v_k \right)^T \cdot A \cdot \left(x_n - v_k \right) \quad (4)$$

La forme des groupes est déterminée par le choix de la matrice A dans l'équation (4). Un choix très commun est $A = I$, ce qui donne la distance euclidienne [BAB98]:

$$D_{knA}^2 = \left(x_n - v_k \right)^T \cdot \left(x_n - v_k \right) \quad (5)$$

Cette distance est la plus utilisée dans l'algorithme des *K-moyennes* et elle induit une partition en hyper-sphères de dimension *D.* Un autre choix, pour A, est une matrice diagonale qui utilise les variances dans les directions des coordonnées de l'espace de représentation. Ceci produit la norme diagonale et génère des classes hyper-ellipsoïdales, parallèles aux axes de l'espace de représentation.

Finalement, A peut être définie comme l'inverse de la matrice de covariance de l'ensemble $\boldsymbol{X}$: $A = M^{-1}$

$$M = \frac{1}{N} \sum_{n=1}^{N} \left(x_n - \bar{x} \right) \cdot \left(x_n - \bar{x} \right)^T \quad (6)$$

Dans l'équation (6), $\bar{x}$ correspond à la valeur moyenne de l'ensemble des éléments *X.* Dans ce cas, la matrice A induit la norme de Mahalanobis dans l'espace *D*. Avec cette norme, les classes sont représentées par des hyper-ellipses dont l'orientation est arbitraire, définie par les vecteurs propres de M, comme le montre la Figure 2.2.

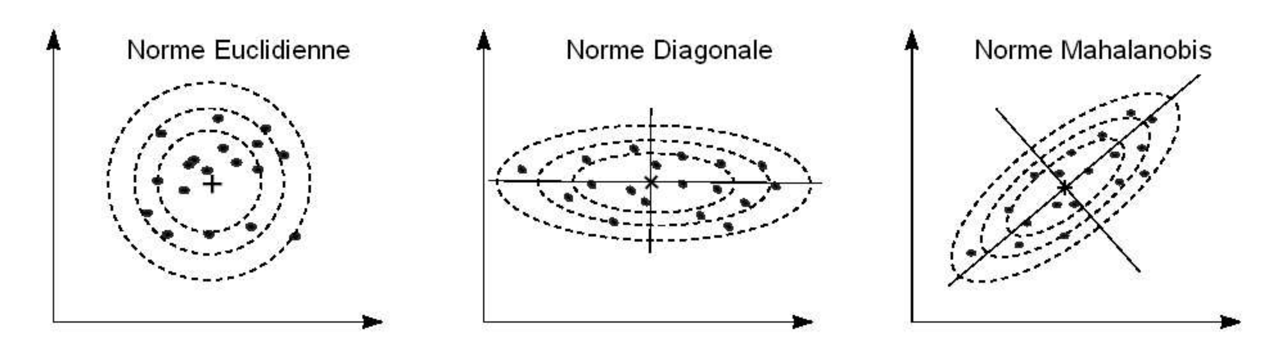


Figure 2.2. Mesures de distance pour le regroupement [BAB98].

L'algorithme simplifié est donné ci-dessous :

Algorithme K-Moyennes *:*

Entrée : K groupes (1<K<N)

1. *Choisir le centre initial des K groupes (aléatoirement)*
2. *Affecter chaque élément* x_n *au groupe dont il est le plus proche*
3. *Recalculer le centre* v_k *de chaque groupe*
4. *répéter 2 et 3 jusqu'à ce que : stabilisation des centres ou nombre d'iterations* t *(fixé) atteint.*

Même s'il peut être démontré que l'algorithme converge toujours, il ne trouve pas forcement la configuration optimale, correspondant au minimum global de la *fonction objectif* donnée par l'équation (3). Ceci correspond à un problème d'optimisation multimodal.

2.3.1.2 C-Moyennes Floues (FCM)

Une extension directe de l'algorithme *K-moyennes* est l'algorithme des ***C-moyennes floues (FCM)***[BEZ81], où la notion d'ensemble flou est introduite dans la définition des classes. Chaque objet a un degré *d'**appartenance*** (ou de validité) associé à chaque classe. De ce fait et comme pour les *K*-moyennes, l'algorithme *FCM* utilise, de manière itérative, un critère J_{FCM} de minimisation des distances intra-classes et de maximisation des distances inter-classes, mais en tenant compte des degrés d'appartenance des éléments :

$$J_{FCM}(X,U,V)=\sum_{k=1}^{K}\sum_{n=1}^{N}(\mu_{kn})^{m}D_{knA}^{2} \tag{7}$$

Où :

- X est la matrice des *N* éléments :

$$\begin{bmatrix} x_1^1 & \cdots & x_1^d & \cdots & x_1^D \\ \vdots & \ddots & & & \vdots \\ x_n^1 & & x_n^d & & x_n^D \\ \vdots & & & \ddots & \vdots \\ x_N^1 & \cdots & x_N^d & \cdots & x_N^D \end{bmatrix}$$

- U est la matrice de partition floue (dimension *K*x*N*) avec $\mu_{kn}\in[0,1]$ qui définit le coefficient d'appartenance de chaque élément x_n à la classe C^k,
- V est la matrice des prototypes v_k ou centres des classes,
- $m\in(1,\infty)$ conditionne le degré flou de la partition (sa valeur est généralement fixée à 2 dans les applications). Le poids de cet exposant s'interprète comme un paramètre de distorsion. Le "flou" de la partition augmente avec ce coefficient. Par conséquent,

les faibles niveaux d'appartenance s'accentuent, car il y a un recouvrement entre les classes.
L'algorithme simplifié est donné ci-dessous :

<u>Algorithme FCM</u> *:*

Entrée: K groupes (classes) où $1 < \boldsymbol{k} < \boldsymbol{N}$ *, degré de flou* $\boldsymbol{m}$ *, métrique* $\boldsymbol{A}$.

Initialiser aléatoirement la matrice d'appartenance $\boldsymbol{U}^{(0)}$.

Répéter pour chaque itération $\boldsymbol{l} = 1,2,\ldots$:

1. *modifier les prototypes (centres) de classes* $\boldsymbol{v_k}^{(l)}$

$$\boldsymbol{v}_{\boldsymbol{k}}^{(\boldsymbol{l})} = \frac{\sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \left(\mu_{\boldsymbol{kn}}^{(\boldsymbol{l}-1)}\right)^{\boldsymbol{m}} \boldsymbol{x}_{\boldsymbol{n}}}{\sum_{\boldsymbol{n}=1}^{\boldsymbol{N}} \left(\mu_{\boldsymbol{kn}}^{(\boldsymbol{l}-1)}\right)^{\boldsymbol{m}}}, \quad 1 \le \boldsymbol{k} \le \boldsymbol{K}$$

2. *calculer les nouvelles distances* $\boldsymbol{D}_{\boldsymbol{kn}}^2$

$$\boldsymbol{D}_{\boldsymbol{knA}}^2 = (\boldsymbol{x}_{\boldsymbol{n}} - \boldsymbol{v}_{\boldsymbol{k}}^{(\boldsymbol{l})})^{\boldsymbol{T}} \boldsymbol{A}(\boldsymbol{x}_{\boldsymbol{n}} - \boldsymbol{v}_{\boldsymbol{k}}^{(\boldsymbol{l})}), \quad 1 \le \boldsymbol{k} \le \boldsymbol{K},\ 1 \le \boldsymbol{n} \le \boldsymbol{N}$$

3. *mettre à jour la matrice* $\boldsymbol{U}^{(l)}$

$$\textbf{\textit{Si}}\ \boldsymbol{D}_{\boldsymbol{knA}} > 0 \quad \textbf{\textit{pour}} \quad 1 \le \boldsymbol{k} \le \boldsymbol{K},\ 1 \le \boldsymbol{n} \le \boldsymbol{N},$$

$$\mu_{\boldsymbol{kn}}^{(\boldsymbol{l})} = \frac{1}{\sum_{\boldsymbol{k}=1}^{\boldsymbol{K}} \left(\boldsymbol{D}_{\boldsymbol{knA}} / \boldsymbol{D}_{\boldsymbol{hnA}}\right)^{2/(\boldsymbol{m}-1)}}$$

$$\textbf{\textit{Sinon}} \quad \mu_{\boldsymbol{nk}}^{(\boldsymbol{l})} = 1\ (\textbf{\textit{distance est 0, appartenance est 1}})$$

Jusqu'à obtenir la stabilité de $\boldsymbol{U}^{(l)}$ *:* $\left\|\boldsymbol{U}^{(l)} - \boldsymbol{U}^{(l-1)}\right\| < \varepsilon$ *, où* ε *correspond au critère de convergence.*

Il existe d'autres extensions de l'algorithme de *FCM,* tel que l'algorithme *GK-Moyennes* de Gustafson et Kessel [GUS79], qui propose une mesure de distance adaptative basée sur la norme de Mahalanobis (équation (6)), pour détecter des classes avec des formes géométriques différentes.

Un désavantage des algorithmes de regroupement ("*clustering")* est qu'ils sont sensibles à la sélection de la partition initiale. Souvent, l'initialisation se fait aléatoirement, plusieurs fois, dans l'espoir qu'une des partitions mène à un regroupement acceptable. Cette sensibilité à l'initialisation devient aiguë quand la distribution des données montre une grande variance.

Les techniques de regroupement sont considérées parmi les méthodes de classification non-supervisées, car elles n'utilisent aucune connaissance a priori des classes qui doivent être créées. Néanmoins, un nombre spécifique de classes doit être choisi à l'avance, peu importe si les classes créées sont significatives ou non, ce qui donne, tout de même, à ces méthodes un caractère supervisé. Ceci peut être un

inconvénient car il n'est pas toujours possible de connaître le nombre de groupes (classes) contenues dans l'ensemble de données.

2.3.2 Méthodes statistiques

Selon Michie *et al.* [MIC94], deux types différents de méthodes sont identifiés comme des procédures statistiques pour la classification. Le premier type correspond aux algorithmes dits *"classiques"* car ils sont dérivés des ***Discriminants Linéaires de Fisher (DL)*** [FIS36]**.** Le deuxième type englobe des techniques de classification *"modernes"* et fournit une estimation de la distribution conjointe des caractéristiques dans chaque classe.

Pour l'approche des *DL* de Fischer, l'ensemble d'apprentissage comprend *N* éléments représentant q classes prédéfinies (connues). L'espace de représentation est défini par *D* attributs de type quantitatif et chaque objet de l'ensemble d'apprentissage est décrit par un vecteur d'attributs $\vec{x} = (x_1, x_2, \ldots, x_D)$. Cet espace de représentation est divisé par un ensemble d'hyperplans, chacun défini par une combinaison linéaire des attributs. L'hyperplan divisant deux classes est tracé de façon à bissecter la ligne joignant les centres de ces classes. La direction de l'hyperplan est déterminée par la forme des groupes des éléments. Cette approche ne requiert pas de suppositions probabilistes. Il y a deux façons de réaliser la classification en utilisant les *DL* :

- Une façon est de maximiser la séparation entre les classes par le critère des *moindres carrés*. Un hyperplan dans l'espace de représentation *D*-dimensionnel est choisi pour séparer du mieux possible les classes prédéfinies. Les éléments sont classés selon le coté de l'hyperplan où ils se situent.
- Une autre possibilité est de faire la classification à l'aide de *l'estimateur maximum de vraisemblance*. Ici, on suppose que les attributs des éléments d'une classe sont indépendants et suivent une certaine distribution de probabilité, avec une fonction de probabilité (fdp) f_i [MIC94].Un nouvel objet, de vecteur d'attributs $\vec{x}$, est alors affecté à la classe pour laquelle la fdp $f_i(\vec{x})$ est maximale.

Ces techniques de classification font l'hypothèse que le vecteur des caractéristiques des objets obéit à une loi de distribution connue. De ce fait, ces méthodes sont appelées aussi *paramétriques* [MAR03][CAS03]. Une supposition très fréquente est que les distributions sont de type *Gaussien (ou normal)*, avec des moyennes différentes mais avec la même matrice de covariance.

La Figure 2.3 illustre les *LD* de Fisher appliqués à la base de données des iris disponibles dans [UCI98]. Il y a trois variétés d'iris : Setosa, Versicolor et Virginica. Il y a 50 mesures pour chaque variété. Nous avons pris deux des quatre attributs pour faire la partition des trois variétés : la longueur et la largeur de pétale de chaque fleur. Ainsi, pour faire la différence entre les Versicolor et les Virginica, les hyperplans ont été construits en utilisant la règle suivante [MIC94]:

SI *Largeur de pétale* < 3.272 – 0.325**Longueur de pétale*, ALORS ***Versicolor***
SI *Largeur de pétale* > 3.272 – 0.325**Longueur de pétale*, ALORS ***Virginica***

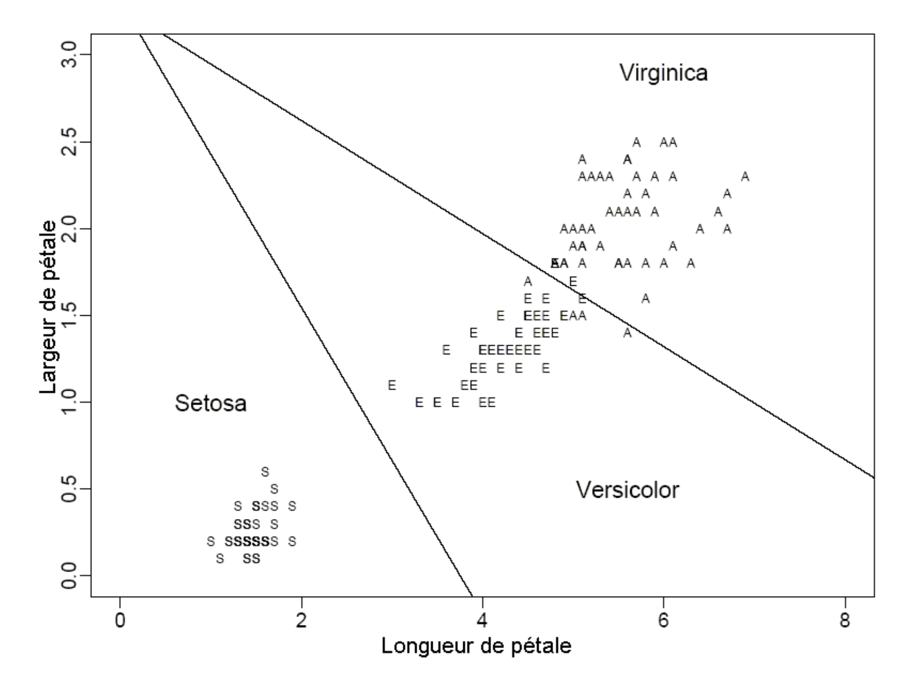


Figure 2.3. Classification par *Discriminants Linéaires* : Données Iris [MIC94]

Les techniques statistiques "modernes" sont considérées comme des procédures de classification non paramétriques qui peuvent être utilisées sans hypothèses sur l'existence d'une loi de probabilité. La méthode des ***K plus proches voisins*** [DUD01a] est une de ces techniques. L'idée générale est : pour un nouvel objet x', il s'agit de déterminer la classe de chacun des *k* plus proches individus de x' parmi tous les objets de l'ensemble d'apprentissage. Nous pouvons alors classer le nouvel objet dans la classe la plus fréquente parmi celles de ses voisins.

Une difficulté de cette technique est de trouver une normalisation appropriée des observations. Pour des données multi-variables, dont les variables sont mesurées selon différentes échelles, une normalisation s'avère nécessaire afin de comparer les individus. Pour des ensembles de données trop importants en nombre, cette technique peut devenir coûteuse en temps (pour *k>1*) puisque toutes les observations de l'ensemble des données d'apprentissage doivent être enregistrées et examinées à chaque nouvelle classification d'un objet.

2.3.3 Arbres de décision et induction

Les **arbres de décision** sont des outils puissants et très répandus pour la classification et la prédiction [MIC94]. Ce qui les rend attractif est le fait qu'ils peuvent être représentés comme des *règles.* Les règles peuvent, aisément, être exprimées de façon interpretable. Dans des applications où l'explication des causes ou des raisons d'une décision est cruciale, les experts ont besoin d'outils qui permettent de reconnaître et de valider les connaissances découvertes issues des données, dans le contexte du

diagnostic. Il existe une variété d'algorithmes pour la construction des arbres de décision qui offrent la qualité d'interprétation souhaitée.

Les arbres de décision sont des classificateurs avec une structure arborescente (voir Figure 2.4). Ils réalisent une recherche *"top-down"* (de haut en bas). Chaque nœud représente une décision, ou un test, à effectuer sur un attribut donné des individus. Le nœud initial s'appelle *racine.* Deux (ou plus) branches peuvent être issues d'un nœud selon si la décision à prendre est de type binaire ou non. Les nœuds intermédiaires sont des *nœuds de décision*. Les nœuds terminaux se nomment *feuilles*, et c'est là que la décision est prise pour affecter un individu à une classe existante. Les conditions principales, pour l'extraction des connaissances à partir de la construction d'arbres de décision, sont les suivantes :

- les classes auxquelles les individus sont assignés doivent être établies à l'avance,
- une quantité suffisante d'exemples (individus) représentatifs des classes est nécessaire,
- pour chaque nœud de décision, l'attribut approprié doit être choisi.

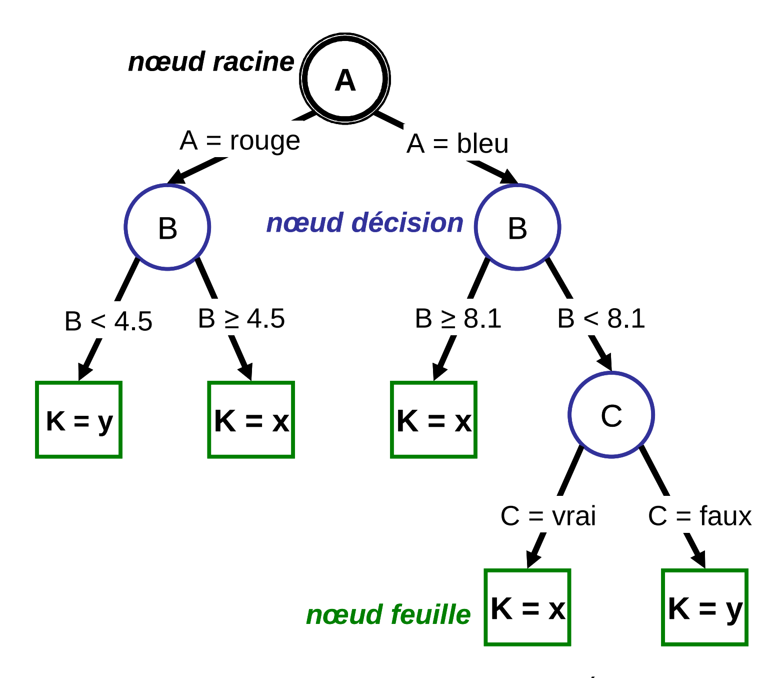


Figure 2.4. Exemple d'un arbre de décision

L'algorithme ID3 (et son successeur C4.5) de J.R Quinlan [QUI86] et l'algorithme CART (*Classification And Regression Tree*) [DUD01] sont les algorithmes les plus connus d'arbres de décision. Le critère d'évaluation dans l'algorithme ID3 est le choix d'un attribut à examiner à chaque noeud de décision. Le but est de choisir, dans tout l'espace de représentation, l'attribut qui sépare le mieux les individus de l'ensemble d'apprentissage. Pour choisir quel attribut est le plus approprié, c'est-à-dire celui avec la distribution de classe la plus non-homogène [MIT97], l'algorithme emploie le concept d'***entropie***, issu de la théorie de l'information. L'entropie caractérise la pureté, ou l'homogénéité, d'une collection d'individus :

$$\boldsymbol{Entropie}(\boldsymbol{A}) = \sum_k - \left(\frac{\boldsymbol{n}_{bk}}{\boldsymbol{n}_b}\right) \log_2 \left(\frac{\boldsymbol{n}_{bk}}{\boldsymbol{n}_b}\right) \quad (8)$$

$$\boldsymbol{Entropie\ Moyenne} = \sum_b \left(\frac{\boldsymbol{n}_b}{\boldsymbol{n}_N}\right) \times \boldsymbol{Entropie} \quad (9)$$

Où :

- n_b est le nombre d'individus dans la branche *b* d'un attribut *A* de l'espace de représentation,
- n_{bk} est le nombre d'individus dans la branche *b* d'un attribut pour la classe *k,*
- n_N est le nombre total d'individus dans toutes les branches.

La procédure à suivre est la suivante : d'abord, il faut trouver la façon dont l'attribut distribue les individus dans les classes selon chaque branche (équation (8)) Ensuite, il faut calculer l'entropie moyenne (équation (9)) de chaque attribut et choisir celui pour lequel l'entropie est moindre; il sera le premier attribut à être testé. La procédure de sélection d'un nouvel attribut et la partition des individus est répétée pour chaque nœud, en n'utilisant que les individus associés au nœud. Ainsi, les attributs qui ont déjà été testés ne sont pas inclus, de façon à ce que chaque attribut soit testé au moins une fois dans un des chemins possibles. Ceci est fait jusqu'à ce que :

- chaque attribut ait été déjà inclus, ou que
- les individus liés à ce nœud feuille aient tous la même classe assignée (i.e. leur entropie est zéro).

Il n'y a aucune restriction pour la nature des données à classifier : les arbres de décision peuvent manipuler des variables continues (discrétisées) et symboliques. Un autre avantage des arbres de décision est leur capacité à générer des règles compréhensibles, tout en réalisant la classification sans exiger beaucoup de calcul. Ils fournissent une indication claire sur les attributs les plus représentatifs pour la classification ou la prévision. Un inconvénient des arbres de décision est que la plupart d'entre eux examinent seulement un champ (attribut) à la fois. D'ailleurs, ils peuvent être informatiquement coûteux dans leur construction, de par leur croissance combinatoire.

2.3.4 Réseaux neuronaux artificiels

Les **réseaux de neurones artificiels (RNA)** constituent une technique non-linéaire complexe basée sur le modèle d'un neurone humain. Il s'agit d'un ensemble d'entités simples, appelées « neurones », fortement interconnectées entre elles. En fait, les RNA visent à modéliser les capacités d'apprentissage du cerveau humain.

Les premiers travaux sur les RNA ont été développés par McCulloch et Pitts en 1943, qui, en se basant sur des fondements biologiques, ont proposé un modèle binaire

où chaque neurone a une fonction seuil. En 1949, Hebb propose une formulation du mécanisme d'apprentissage, sous la forme d'une règle de modifications des connexions synaptiques. Ce mécanisme est devenu la base des algorithmes courants.

La première architecture neuronale, le ***perceptron***, apparaît en 1958, grâce aux travaux de Rosenblatt [PER98].

Un RNA se compose typiquement d'une couche d'entrée, d'une ou de plusieurs couches intermédiaires ou *cachées* et d'une couche de sortie. Dans le fonctionnement du RNA, chaque nœud ou neurone est connecté à d'autres neurones d'une couche suivante par des poids synaptiques adaptables.

La Figure 2.5 montre le schéma général d'un neurone. Chaque nœud *i* calcule la somme de ses entrées (x_1,…,x_N), pondérées par les poids synaptiques (w_{i1},…,w_{iN}) correspondants; cette valeur représente l'*état interne du neurone* (u_i). Ce résultat est alors transmis à une fonction d'activation f (voir Figure 2.6 pour la représentation des fonctions d'activation les plus utilisées). La sortie y_i est l'activation du neurone.

Le modèle général du neurone est représenté par :

$$u_i = \sum_{n=1}^{N} w_{in} x_n \quad (10)$$

$$y_i = f(u_i) \quad (11)$$

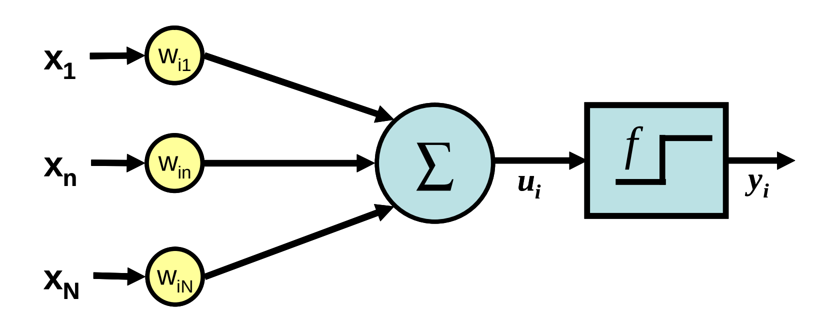


Figure 2.5. Schéma général d'un neurone

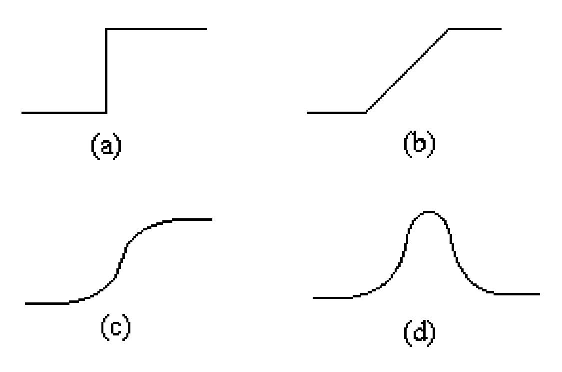


Figure 2.6. Fonctions d'activation les plus utilisées : (a) fonction à seuil, (b) fonction linéaire, (c) fonction sigmoïde, (d) fonction gaussienne

La propriété principale des RNA est leur capacité d'apprentissage. En fait, cet apprentissage peut être considéré comme la mise à jour des poids des connexions au sein du réseau. Il existe différentes techniques d'apprentissage, dont la plupart ont besoin d'une quantité considérable d'exemples et de beaucoup de temps de calcul. Une de ces techniques est l'***apprentissage supervisé***; ici le réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie. De cette façon, le réseau va modifier ses poids jusqu'à ce que le résultat soit le plus proche possible de la sortie attendue, correspondant à une entrée donnée. L'algorithme de ***rétro-propagation*** (Figure 2.7) est le plus populaire parmi les techniques d'apprentissage supervisé [BIS95]. L'algorithme essaye d'améliorer la performance du RNA en réduisant la différence entre le résultat (y_i) obtenu par le nœud et la sortie attendue (d_i), en ajustant les poids par une téchnique de gradient. Cette différence est mesurée par l'erreur quadratique :

$$E = \frac{1}{2}\sum\left(y_i - d_i\right)^2 \tag{12}$$

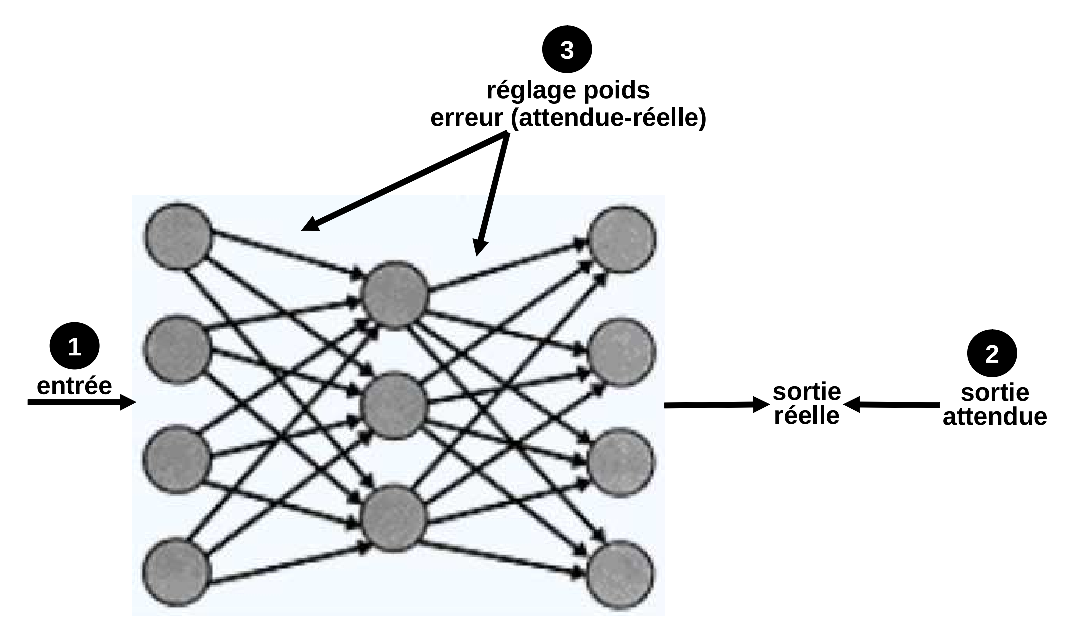


Figure 2.7. Réseaux avec apprentissage par retro-propagation

L'apprentissage par rétro-propagation travaille de la façon suivante : les poids w_{in} son initialisés aléatoirement. Chaque fois qu'un exemple est présenté au RNA, l'activation de chaque nœud est calculée. Après que le résultat de la sortie soit calculé, l'erreur δ_{ip} de chaque nœud est calculée en remontant le réseau, c'est-à-dire de la couche de sortie vers la couche d'entrée. Cette erreur est le produit de la fonction d'erreur E (équation (12)) et de la dérivée de la fonction d'activation f. Par conséquent, l'erreur est une mesure du changement de la sortie du RNA provoqué par un changement des valeurs des poids du réseau. Les RNA avec apprentissage supervisé les plus utilisés sont le perceptron, le perceptron multi-couche (PMC) [HER94] et les réseaux à base radiale (RBF) [ORR96]. Les RBF permettent de réaliser un

apprentissage sans déterminer *a priori* la structure du réseau. La partition de l'espace de description selon les RBF est décrite par des formes géométriques diverses. Les RBF sont des réseaux à deux couches. Dans la première couche, des fonctions à *Base Radiale* sont implémentées au sein des neurones. Normalement la fonction utilisée est de type gaussien. Les poids de cette couche, sont calculés en utilisant la fonction de distance euclidienne. La deuxième couche (couche de sortie) correspond aux neurones contenant une fonction d'activation de type linéaire.

Dans la plupart des algorithmes utilisés pour la conception des RNA, il faut choisir à l'avance le nombre de couches cachées du réseau ainsi que le nombre de neurones par couche.

Dans le cas des RNA avec ***apprentissage non-supervisé***, où les exemples n'ont pas des classes attribuées *a priori*, le réseau s'adapte en réponse à ses entrées. Il y a des réseaux capables de modéliser la structure de l'espace de représentation, généralement en termes d'une fonction de densité de probabilité, ou en représentant les données en termes de prototypes (centres) et leur similarité à chaque prototype. Cette similarité est calculée avec une métrique de distance. Quand la mesure de similarité d'un exemple est considérée insuffisante pour que celui-ci soit classé dans un des différents groupes existants, un nouveau groupe est créé. Parmi ce type de réseau, nous trouvons les réseaux auto-organisationnels (ou de Kohonen) [KOH95] et plus récemment, les réseaux "*Cluster Détection and Labeling (CDL)*" [ELT98].

Après la phase d'apprentissage à partir d'un ensemble des données, le RNA est prêt et peut être utilisé pour la prédiction ou pour la classification de nouveaux exemples dans les classes existantes.

Dans la littérature, de nombreux travaux démontrent l'utilité des RNA, de par leurs caractéristiques d'apprentissage et capacité de généralisation, pour la détection et le diagnostic de défaillances [VEN90], ainsi que pour la surveillance de processus complexes [LUR03].

Les RNA sont très intéressants pour des problèmes non-linéaires où il devient de plus en plus difficile d'employer des techniques telles que les arbres de décision, car ces derniers coupent l'espace de représentation de façon parallèle aux axes (attributs). Les RNA sont appropriés aussi dans le cas de mesures bruitées. Néanmoins, le temps de calcul pendant la phase d'apprentissage peut devenir très important et la définition des paramètres du réseau délicate. Un autre inconvénient est que les RNA ne donnent pas une représentation explicite des connaissances, ni à partir des règles, ni avec un autre moyen facilement interprétable. Le modèle est implicite et caché dans la structure du réseau et les poids des connexions entre nœuds. C'est pour cette raison que les RNA sont parfois considérés comme "un système boîte grise". En outre, les RNA ne permettent pas de travailler directement avec des informations de type symbolique (qualitatif). Donc, dans un problème de classification où certains attributs sont de type qualitatif il est nécessaire de réaliser une représentation quantitative. Néanmoins, Nauck et Kruse ont proposé une approche neuro-floue pour la classification de données [NAU95]. Il s'agit, de réaliser une classification à partir des données dont leurs attributs sont représentés par des sous-ensembles flous.

2.3.5 LAMDA

LAMDA (*Learning Algorithm for Multivariable Data Analysis*) est une méthode de classification avec apprentissage, proposée par Joseph Aguilar-Martin, en collaboration avec plusieurs chercheurs (López de Mántaras, Piera, …)[AGUI82][DES87][PIE89]. Depuis les années 1980, de nombreuses améliorations ont été apportées à la méthode. Plus récemment, une description détaillée de cette méthode a été présentée dans les travaux de thèse de doctorat de Aguado [AGUA98] et Waissaman-Vilanova [WAI00]. LAMDA a été utilisée dans des domaines très divers tels que la psychologie [GAL02], les procédés biotechnologiques [WAI99] ainsi que les procédés industriels [KEM03].

Une des motivations, à l'origine du développement de cette méthode, est la volonté d'appréhender les difficultés inhérentes au diagnostic à base de données historiques. En effet :

- Plus on considère d'attributs pour la description d'un élément, plus il est probable de trouver cet élément différent des autres. Ceci peut amener à une classification avec un nombre excessif de classes, dont la plupart avec un seul individu.
- Certains attributs caractérisant les individus sont représentés par des valeurs imprécises ou approximatives, ou bien correspondent à des situations qualitatives pas forcement binaires, c'est-à-dire que leur présence ou leur absence pour un élément donné est connue avec un certain degré d'incertitude.

Pour ces raisons, l'organisation d'un ensemble d'individus en groupes ou classes doit aller au delà de l'idée de faire une partition logique stricte, comme il est déjà fait dans la méthode des *C-moyennes floues* (voir § 2.3.1.2).

LAMDA est une méthode de classification et de regroupement basée sur la logique floue. L'idée générale est de trouver le degré d'***adéquation*** *(appartenance)* d'un élément (ou individu) à des classes existantes. Cette adéquation est obtenue à partir de l'analyse des contributions de chacun des attributs de l'individu aux caractéristiques des classes. Cette contribution est appelée le ***degré d'adéquation marginal (DAM)***. Une fois que tous les DAMs sont calculés le ***degré d'adéquation globale (DAG)*** d'un individu à une classe donnée est déterminé au moyen des opérateurs logiques d'agrégation appelés *connectifs mixtes de compensation linéaire*, introduits et étudiés en [PIE91].

LAMDA offre la possibilité d'effectuer un apprentissage supervisé (avec professeur) et/ou non-supervisé (auto-apprentissage). Cet apprentissage est fait de façon incrémentale et séquentielle; les données sont traitées une fois, ce qui permet de réduire la phase d'apprentissage à une ou très peu d'itérations. Quoi qu'il en soit, l'attribution d'un individu à une classe suit la même procédure, qu'il soit le premier à initialiser une classe ou qu'il contribue à la modification d'une classe existante. La Figure 2.8 illustre la procédure d'attribution d'un individu à une classe.

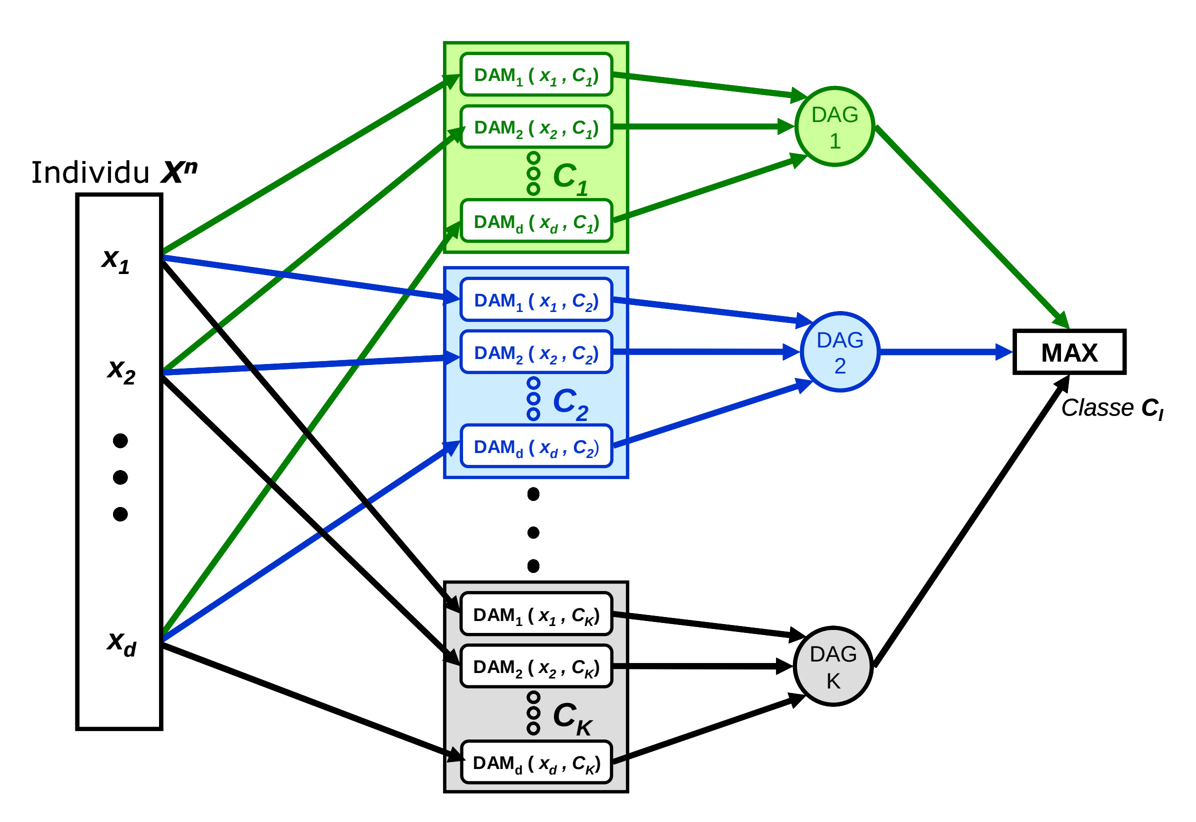


Figure 2.8. Affectation d'un individu à une classe

A. *Degré d'Adéquation Marginale (DAM)*

De façon générale, chaque individu est représenté par un vecteur X avec *D* descripteurs ou attributs $\{x_1,\ldots,x_D\}$. Ces descripteurs *e.g. poids, température, etc.*, peuvent avoir des valeurs quantitatives, mais il existe aussi des descripteurs tels que la *couleur, forme, etc.* dont les valeurs sont de type qualitatif (symbolique) et appelées ***modalités***. Pour le cas des descripteurs quantitatifs, les valeurs observées suivent une normalisation par rapport à la valeur maximale x_{jMAX} et la valeur minimale x_{jMIN} que le descripteur *j* peut avoir, de manière à ce que ces valeurs se trouvent toujours dans l'intervalle [0,1]. Cette normalisation est donnée par :

$$x_j = \frac{\hat{x}_j - x_{jMIN}}{x_{jMAX} - x_{jMIN}} \quad (13)$$

Où, pour le descripteur *j* d'un élément, $\hat{x}_j$ est la valeur brute et x_j est la valeur normalisée.

L'individu doit être attribué à une des *K* classes existantes. Pour effectuer ceci, pour chaque classe, *D* adéquations marginales doivent être calculées ($DAM_1(x_1, C_k),\ldots, DAM_j(x_j, C_k),\ldots DAM_D(x_D, C_k)$). Il est bien entendu que les paramètres des classes ont les mêmes caractéristiques que les individus pour pouvoir calculer les DAMs. Puisque LAMDA permet de classer des individus représentés par des descripteurs quantitatifs ainsi que qualitatifs, le calcul des DAMs est réalisé différemment selon le cas:

- **DAM Quantitatif :** Dans le cas où le descripteur est de type quantitatif, il existe plusieurs fonctions qui ont été proposées pour le calcul des DAMs. Nous citerons ici trois fonctions qui ont été utilisées dans notre travail.

1. Fonction Binomiale Floue

Il s'agit d'une extension floue de la fonction *binomiale* [AGUI80]. Cette fonction est une estimation Bayesienne, non-parametrique, c'est-à-dire, il n'y a pas de distribution probabiliste imposée. Étant basée sur une loi de probabilité binômiale, elle exprime intrinsèquement le degré entre l'adéquation et l'inadéquation :

$$\boldsymbol{DAM}(\boldsymbol{x_j},\rho_{\boldsymbol{kj}}) = \rho_{\boldsymbol{kj}}{}^{x_j}(1-\rho_{\boldsymbol{kj}})^{1-x_j} \tag{14}$$

Dans l'équation (14), pour un descripteur quantitatif *j* : ρ_{kj} est la valeur du paramètre de la classe C_k qui décrit le descripteur et x_j correspond à la valeur normalisée observée dans l'individu. Cette fonction fait une partition de l'espace de représentation par rapport aux valeurs extrêmes, c'est-à-dire qu'il y a une tendance vers la bipolarisation "réussite-échec", identifiées à 1 et 0. (voir Figure 2.9).

2. Fonction Binomiale Centrée

Waissman [WAI98] a proposé une fonction qui permet une partition autour des centres des classes. Le DAM est calculé alors par la proximité entre la valeur x_j observée pour le descripteur *j* et le centre c_{kj} du même paramètre pour la classe *k* :

$$\boldsymbol{DAM}(\boldsymbol{x_j},\rho_{\boldsymbol{kj}}) = \rho_{\boldsymbol{kj}}{}^{1-d_{kj}}(1-\rho_{\boldsymbol{kj}})^{d_{kj}} \tag{15}$$

où la distance est : $$d_{kj} = \left|x_j - c_{kj}\right| \tag{16}$$

3. Fonction Gaussienne

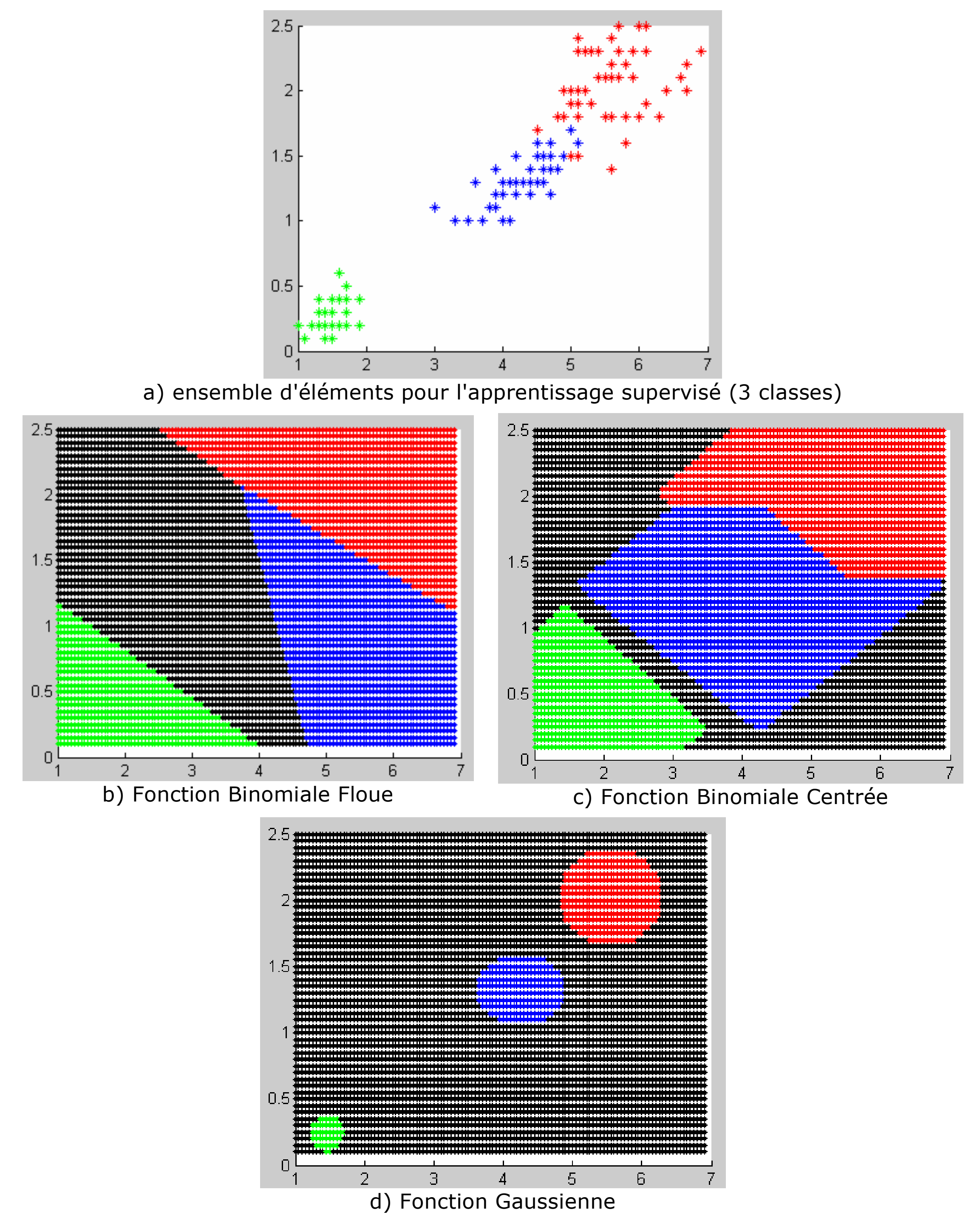
Finalement, dans les cas où la distribution du descripteur peut être considérée comme une distribution de type normal, il est possible d'utiliser une fonction de type gaussienne :

$$\boldsymbol{DAM}(\boldsymbol{x_j},\boldsymbol{C_{kj}}) = \boldsymbol{e}^{-\frac{1}{2\sigma_{kj}^2}(x_j-\mu_{kj})^2} \tag{17}$$

où μ_{kj} et σ_{kj}^2 correspondent, respectivement, à la valeur moyenne et à la variance du descripteur *j* pour la classe *k*. Avec cette fonction, les classes ont une forme hyper-ellipsoïdale orientée, parallèle aux axes de l'espace de représentation (voir Figure 2.9). Il s'agit d'une distance de type diagonale.

La Figure 2.9 montre l'influence du type de fonction pour le calcul du DAM. Ceci permet l'adaptation de la méthode à la distribution attendue des données utilisées pour

la classification. Ainsi, dans le cas de la Figure 2.9 où les individus sont caractérisés par deux descripteurs, avec la fonction *binomiale floue,* les classes sont séparées par des droites et le groupement se fait vers les extrémités; dans le cas de la fonction *binomiale centrée,* les classes sont également séparées par des morceaux de droites mais cette fois autour d'un centre; finalement, pour la fonction du type gaussien, le regroupement donne des ellipses orientées parallèlement aux axes des descripteurs.



a) ensemble d'éléments pour l'apprentissage supervisé (3 classes)

b) Fonction Binomiale Floue

c) Fonction Binomiale Centrée

d) Fonction Gaussienne

Figure 2.9. Partition de l'espace de description selon les fonctions DAM pour le cas quantitatif.

- **DAM Qualitatif:** pour un individu ***X***, la valeur d'un attribut qualitatif peut être une de ses *m* modalités. Chaque modalité est associée à une valeur π_i qui indique la fréquence de celle-ci dans les objets appartenant à la classe C_k. L'ensemble des *m* fréquences observées dans la classe C_k, pour chaque modalité, est le vecteur $\Psi_{kj} = (\pi_1,...,\pi_i,...,\pi_m)$. Pour le calcul du DAM, la présence de la modalité *i* désigne un masque qui recouvre toutes les autres modalités et seulement la fréquence de la modalité correspondante sera prise en compte :

$$DAM(x_j, \Psi_{kj}) = \pi_i \quad (18)$$

B. Degré d'Adéquation Globale (DAG)

Une fois que les DAMs pour une classe ont été obtenus, le degré d'adéquation global (*DAG*) doit être calculé. Comme nous l'avons mentionné précédemment, le DAG est obtenu par l'agrégation des DAMs en utilisant des connectifs mixtes [PIE91]. Ces connectifs sont issus de la théorie des ensembles flous. Ils effectuent une interpolation entre l'opérateur logique d'intersection (T-norme) et celui de l'union (T-conorme), par le biais du paramètre α, par la formule :

$$DAG_\alpha(DAM_1,..., DAM_d) = \alpha T(DAM_1,..., DAM_d) + (1-\alpha)S(DAM_1,..., DAM_d) \quad (19)$$

Le paramètre α est appelé ***indice d'exigence*** et $\alpha \in [0,1]$, *T* correspond à la T-norme (intersection) et *S* correspond à la T-conorme (union). Ainsi, si la valeur zéro est choisie pour α, la classification est peu exigeante dans l'attribution d'un individu à une classe. L'exigence la plus forte est obtenue pour $\alpha = 1$. Différentes familles de connectifs flous peuvent être choisies pour le calcul des DAG. En procédant de cette façon, pour un individu, le DAG pour toutes les classes est calculé. L'individu est affecté à la classe dont le degré d'appartenance globale correspondant est maximum. S'il y en a plusieurs, c'est la première classe trouvée qui est choisie.

Pour éviter l'attribution d'un individu peu représentatif à une classe, c'est-à-dire un individu avec une faible appartenance à toutes les classes, un seuil minimum pour le DAG est employé. Ce seuil correspond au DAG obtenu pour la *Classe Non-Informative (NIC)*. Cette classe a la particularité d'accepter tous les individus de l'espace de description avec le même DAG. Elle représente l'homogénéité totale de l'espace de description. Par conséquent, si la valeur maximale des DAGs pour un individu est inférieure au seuil NIC, cet individu ne sera attribué à aucune classe existante. La classe NIC est obtenue en considérant $\rho_{NIC,j} = \frac{1}{2}$ pour tous les descripteurs quantitatifs (dans le cas des fonctions DAM des équations (14) et (15), et pour le cas de l'équation (17) un seuil doit être défini) et le vecteur $\psi_{NIC,j} = \left(\frac{1}{m},...,\frac{1}{m},...,\frac{1}{m}\right)$ pour les descripteurs qualitatifs. La classe NIC reste toujours vide en auto-apprentissage puisqu'une nouvelle classe dont les paramètres sont fonction des valeurs de l'élément est créée. Piera et

Aguilar [PIE91] ont montré que, dans le cas de reconnaissance sans apprentissage, le nombre d'éléments attribués à la classe NIC croît avec l'exigence.

Nous avons mentionné, précédemment, qu'avec LAMDA un auto-apprentissage (apprentissage non supervisé) ainsi qu'un apprentissage dirigé (supervisé) étaient possibles. Pour le premier, aucune connaissance *a priori* sur les classes ou sur leur nombre n'est nécessaire. En ce qui concerne l'apprentissage dirigé, les paramètres nécessaires pour la caractérisation des classes sont les vecteurs des valeurs moyennes pour les descripteurs quantitatifs et/ou les fréquences des modalités pour les descripteurs qualitatifs. L'apprentissage dirigé consiste à calculer la valeur moyenne (équation (20)) pour chaque descripteur ainsi que les fréquences observées à partir de l'ensemble d'individus pré-affectés à la classe C_k.



$$\rho_{ki} = \frac{1}{N_k} \sum_{x \in C_k} x_i \qquad (20)$$

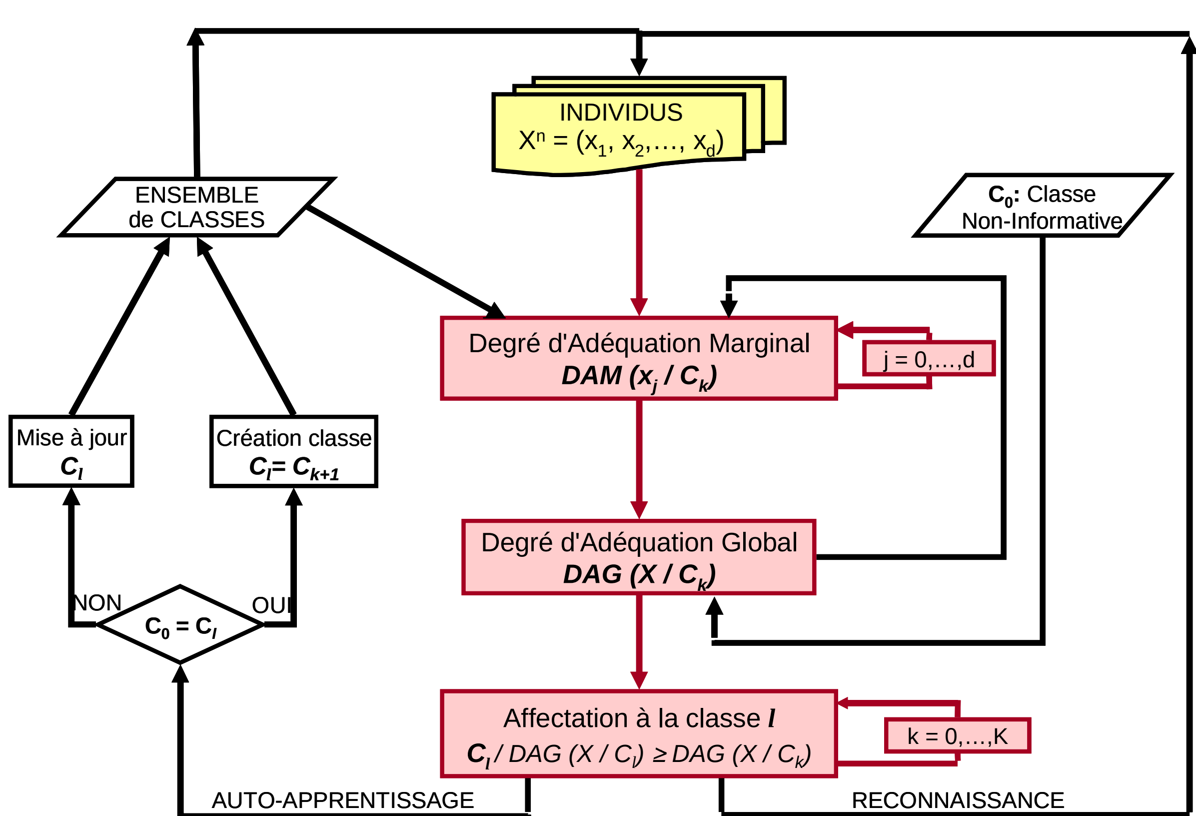


Figure 2.10. Algorithme général de LAMDA

Le schéma de la Figure 2.10 illustre l'algorithme LAMDA dans le cas de l'auto-apprentissage ou bien dans le cas de la reconnaissance. Dans le cas de l'apprentissage, sachant qu'il s'effectue de façon séquentielle, la représentation d'une classe varie après qu'un élément ait été attribué. La mise à jour de la classe s'effectue en prenant en compte les caractéristiques du nouvel élément ainsi que la description de la classe à l'instant précédent.

C. Mise à jour des classes pendant l'apprentissage

En ce qui concerne les descripteurs quantitatifs, les paramètres à mettre à jour dépendent de la fonction choisie pour le calcul des DAMs :

1. Ainsi, pour la fonction **binomiale floue**, nous avons *l'algorithme itératif de la moyenne* :

$$\hat{\rho}_{kj} = \rho_{kj} + \frac{x_j - \rho_{kj}}{N+1} \tag{21}$$

2. Dans le cas de la fonction **binomiale centrée**, deux paramètres doivent être estimés de façon récursive : le nouveau centre $\hat{c}_{kj}$ et la nouvelle probabilité $\hat{\rho}_{kj}$ par le biais de la nouvelle distance $\hat{d}_{kj}$. Ici aussi pour calculer ces deux paramètres, l'algorithme précédent peut être utilisé :

$$\hat{c}_{kj} = c_{kj} + \frac{x_j - c_{kj}}{N+1} \tag{22}$$

$$\hat{d}_{kj} = \left| x_j - c'_{kj} \right| \tag{23}$$

$$\hat{\rho}_{kj} = \rho_{kj} + \frac{1 - d'_{kj} - \rho_{kj}}{N+1} \tag{24}$$

3. Dans le cas d'une fonction de distribution normale, nous devons calculer la nouvelle moyenne $\hat{\mu}_{kj}$ et la variance $\hat{\sigma}^2_{kj}$:

$$\hat{\mu}_{kj} = \mu_{kj} + \frac{x_j - \mu_{kj}}{N+1} \tag{25}$$

$$\hat{\sigma}^2_{kj} = \sigma^2_{kj} + \frac{\left(x_j - \mu_{kj}\right)^2 - \sigma^2_{kj}}{N+1} \tag{26}$$

Dans tous les cas, ***N*** représente le nombre d'éléments attribués précédemment à la classe.

4. En ce qui concerne la mise à jour des descripteurs qualitatifs, la fréquence observée π_i de la modalité *i* est la moyenne de la fonction d'occurrence δ_i :

$$\delta_i = \begin{cases} 1 & \textit{si modalité présente} \\ 0 & \textit{autrement} \end{cases} \tag{27}$$

$$\hat{\pi}_i = \pi_i + \frac{\delta_i - \pi_i}{N+1} \tag{28}$$

De cette façon, la somme de toutes les fréquences d'un descripteur est égale à 1.

Dans le cas de l'auto-apprentissage, quand la valeur maximale obtenue des DAGs d'un individu ne dépasse pas le seuil minimum de la NIC, la création d'une nouvelle classe est possible. Cette nouvelle classe est construite par la modification des caractéristiques de la NIC à partir de celles de l'élément non reconnu. Pour permettre une évolution de la nouvelle classe, la procédure considère que la NIC contient un nombre N_0 fictif d'éléments. Si $N_0 = 0$, la nouvelle classe serait trop figée sur son premier élément et l'affectation, par la suite, d'autres éléments serait peu probable. Une fois la nouvelle classe créée, elle est ajoutée à l'ensemble des classes existantes et, pour les éléments suivants, le DAG à cette classe sera intégré dans la recherche du maximum.

2.3.6 Récapitulatif pour le choix d'une méthode de classification

Afin de montrer une vision générale des méthodes que nous avons citées précédemment, nous présentons dans le Tableau 2.1 les caractéristiques qui sont présentes ou absentes dans chaque méthode.

À partir des caractéristiques du tableau ainsi que des descriptions des méthodes énoncées précédemment, nous avons choisi la méthode LAMDA pour la mise en place de la stratégie de détection et de diagnostic proposée (§2.2).

Ce choix a été fait en vue de la possibilité d'utiliser des attributs de natures différentes (quantitatifs et qualitatifs simultanément) sans avoir besoin de faire une transformation des caractéristiques symboliques (pas forcement binaires). Ceci s'avère avantageux en offrant la possibilité d'utiliser des informations imprécises ainsi que des représentations linguistiques faites par les experts.

Par ailleurs, en comparant LAMDA avec les RNA, il est possible de remarquer une ressemblance au niveau de la structure, qui s'apparente à celle d'un neurone. Néanmoins, LAMDA ne cherche pas à minimiser des fonctions, l'algorithme n'est pas non plus le résultat des combinaisons de poids "inconnus", mais une modification cohérente des caractéristiques des classes.

En plus, le fait d'utiliser les concepts de la logique floue permet une approximation du raisonnement humain qui conduit à des résultats facilement interprétables.

En outre, l'algorithme ne demande pas de faire des initialisations aléatoires et, pour certains choix des fonctions DAM, l'utilisateur ne doit pas faire le choix du seuil minimum qu'un individu doit franchir pour être attribué à une classe. Ce seuil est donné automatiquement par la classe NIC en fonction de l'espace de description.

La possibilité d'ajuster l'exigence peut être vue comme une façon de généraliser ou de donner un degré de "flou" à la classification, permettant d'obtenir différentes partitions pour chaque valeur d'exigence du même ensemble de données [PIE91]. Aussi, même si le résultat final de la classification est présenté comme une partition classique (concrète), le fait d'obtenir les différentes appartenances d'un individu à toutes les classes permet de faire une analyse postérieure sur le voisinage des classes ainsi que sur les individus.

Une autre raison pour laquelle nous avons choisi LAMDA parmi d'autres méthodes est le traitement séquentiel des données. Contrairement aux méthodes statistiques ou de "clustering", l'algorithme n'a pas besoin de garder l'ensemble des données, mais uniquement les paramètres des classes. LAMDA traite chaque élément de façon indépendante. Ceci veut dire que, au moins dans la phase d'apprentissage, LAMDA est plus rapide et sa complexité de calcul est moindre. Par contre, un inconvénient possible de ceci est la dépendance à l'ordre dans lequel les données sont fournies en mode d'auto-apprentissage. Cependant, pour le type de procédés industriels auxquels nous nous intéressons, ce dernier point n'est pas un obstacle dans la mesure où les données que nous considérons sont des mesures réalisées en ligne et, donc, selon un ordre chronologique.

Finalement, LAMDA permet de combiner, l'auto-apprentissage avec l'apprentissage dirigé, c'est-à-dire qu'il est possible de reajuster les paramètres des classes et, donc, de les faire évoluer au fur et à mesure que des nouveaux éléments sont présentés. Ceci admet l'utilisation de plusieurs jeux de données pour la caractérisation des états fonctionnels du processus, permettant, ainsi, d'avoir un modèle de référence plus complet.

Tableau 2.1.

Récapitulatif des principales caractéristiques des méthodes de classification

	Discriminant Linéaire	*K*-Voisins	Clustering GK-moyennes	Arbres de Décision CART	RNA RBF	LAMDA
Paramétrique	oui	non	non	non	non	non
Nature de données	numérique	numérique	numérique	numérique & symbolique	numérique	numérique & symbolique
Besoin d'un ensemble connu d'apprentissage	oui	oui	oui	oui	oui	oui/non
Classes attribuées a priori	oui	oui	non	oui	oui	oui/non
Nombre fixe de classes initiales	oui	oui	oui	oui	non	non
Paramètres à régler	---	K	K, m, ε	---	valeur d'arrêt fonction d'activation de la couche cachée	Fonction DAM connectifs α
Type de classification	strict	strict	flou	strict	strict	strict & flou
Mise à jour des classes	non	non	non	non	non	oui

*oui/non : 2 options sont possibles

2.4 Conclusion

Dans ce chapitre, nous avons présenté une stratégie de détection et diagnostic au moyen des méthodes de classification. Dans un premier temps, nous avons décrit les étapes à suivre pour construire un modèle de référence sur le comportement du processus. Ce modèle est élaboré à partir de l'analyse des informations provenant des mesures de capteurs, historiques et en ligne, ainsi que des connaissances expertes. Un premier modèle est obtenu à partir des résultats issus d'une phase d'apprentissage. Puis, une étape de suivi du procédé peut être réalisée. Ce suivi consiste à l'identification en ligne des états attendus. La détection d'une déviation, à travers la non-reconnaissance de certaines observations, met en évidence un symptôme d'une situation non prévue. Alors, la caractérisation du symptôme à l'aide des connaissances de l'expert permet de savoir s'il s'agit d'une défaillance ou non. Cette démarche offre la possibilité de compléter ainsi le modèle de référence.

Deuxièmement, nous avons présenté les caractéristiques générales des différentes méthodes de classification qui peuvent être utilisées pour la détection de défaillances. Parmi ces méthodes, nous avons évoqué les méthodes de regroupement ("clustering") strict (*K-moyennes)* et flou (*C-moyennes floues)*, lesquelles utilisent un apprentissage de type non supervisé. Parmi les méthodes qui emploient un apprentissage supervisé (ou dirigé), nous avons cité : les arbres de décision, qui peuvent être représentés comme des règles interprétables par l'homme et qui admettent des attributs de nature qualitative ainsi que numérique; les méthodes statistiques et les réseaux des neurones artificiels (RNA). Enfin, nous avons décrit la méthode LAMDA qui repose sur le concept d'adéquation d'un objet aux classes existantes. Les bases théoriques de la méthode se trouvent dans le domaine de la logique floue, même si pour certains cas concrets, elle est équivalente à une méthode statistique. LAMDA réalise la classification d'éléments ayant des caractéristiques quantitatives, qualitatives ou mixtes. Avec LAMDA, un auto-apprentissage ainsi qu'un apprentissage dirigé sont possibles.

Nous avons fait une comparaison des avantages et faiblesses de ces méthodes de manière à sélectionner celle sur laquelle va s'appuyer la stratégie de surveillance que nous proposons.

Dans le chapitre 3, nous allons approfondir le problème du paramétrage nécessaire pour mettre en place un système de surveillance basé sur la classification. Nous détaillerons les éléments à prendre en compte mais, aussi, nous évoquerons les prestations que doivent apporter les outils de supervision pour aider l'opérateur humain dans la prise de décision.

Chapitre 3

3. IDENTIFICATION DES ETATS FONCTIONNELS POUR LA SURVEILLANCE DES PROCEDES A PARTIR DE DONNEES

3.1 Introduction

Après avoir détaillé les différentes étapes à suivre pour l'élaboration d'un système de surveillance à partir de méthodes de classification, il est nécessaire de décrire la phase essentielle : *l'apprentissage*. Cette phase correspond à la caractérisation du comportement du processus. En effet, il s'agit de la construction, à partir des observations issues du processus, d'un classificateur caractérisé par la structure des classes qui permettent d'identifier un ensemble de situations. Ceci correspond, alors, à l'obtention d'un *modèle de comportement*. Le paramétrage du classificateur est obtenu en considérant des aspects tels que : le type et la quantité d'attributs dans l'espace de description, le réglage des paramètres de la méthode de classification sélectionnée, le type d'apprentissage (dirigé ou non) selon les connaissances *a priori* sur les situations caractérisées par les observations ou même leurs appartenances à ces situations. Une fois le classificateur construit, le suivi des situations attendues, identifié comme étant la *phase de reconnaissance*, vise à associer toute nouvelle observation à l'une des classes déterminées.

Il est évident que l'étape du suivi repose essentiellement sur le résultat de la phase d'apprentissage, ce qui pose le problème de l'exhaustivité et de la validité du modèle de comportement. En effet :

- il est fort probable que toutes les classes ne soient pas connues *a priori;* l'expert peut avoir une certaine connaissance sur quelques situations représentées par certaines observations de l'ensemble d'apprentissage, mais il peut s'avérer difficile d'en identifier d'autres,

- il est possible que les classes ne soient pas correctement représentées par les observations disponibles initialement,
- il se peut que, pour certaines situations possibles, l'acquisition des données soit difficile ; il s'agit, souvent, de situations critiques qui ne se produisent pas fréquemment.

Dans ces conditions, rien ne permet d'assurer l'exhaustivité de la connaissance du comportement du processus car les données "historiques" ne peuvent pas couvrir la vie entière du processus. En plus, au fur et à mesure de la présentation de nouvelles observations, des nouvelles situations peuvent apparaître dans la structure initiale. De même, les classes existantes peuvent aussi évoluer au cours du temps. Pour ces raisons, il est nécessaire que le système de surveillance présente un caractère ***adaptatif***, non seulement au moment de la construction du modèle de comportement (phase d'apprentissage) mais, aussi, au moment de l'identification des nouvelles situations. Pour ceci, deux principes d'apprentissage sont prévus : un *apprentissage hors ligne* et un *apprentissage en ligne*. L'apprentissage hors ligne est utilisé, bien sûr, dans l'étape de caractérisation des situations, mais aussi pour essayer de caractériser des nouvelles situations qui peuvent se présenter lorsqu’un nombre suffisant d'observations n'est plus reconnu, compte tenu des classes existantes. L'apprentissage en ligne permet de modifier en continu les classes existantes et, même, d'en créer des nouvelles, tout au long de l'étape du suivi.

Ainsi, la première partie de ce chapitre est consacrée à la présentation des paramètres à régler pendant l'élaboration du modèle de comportement du processus. Nous insisterons, plus particulièrement, sur les paramètres relatifs à la méthode LAMDA. La deuxième partie est dédiée à l’aide pour l'analyse et l'interprétation des résultats, de façon à obtenir un classificateur convenable pour identifier par la suite les situations attendues. Finalement, nous présenterons l'étape de reconnaissance des situations en ligne et les démarches à suivre dans le cas de la détection des déviations de comportement.

3.2 Éléments de paramétrage du système de surveillance

Le premier aspect à considérer dans l'élaboration du système de surveillance est le paramétrage du classificateur, c'est-à-dire les réglages nécessaires pour l'obtention du **modèle de référence** (comportement) à partir d’informations issues directement des mesures du processus ou bien par le biais d'outils de filtrage ou d'extraction d'information, et de la méthode de classification sélectionnée.

Ce paramétrage est fait hors ligne et nécessite une intervention active de l'expert, de façon à intégrer ses connaissances dans toutes les étapes. Pour cela, nous devons fournir à l'opérateur expert des outils lui permettant de faire ce réglage en lui donnant des critères de choix objectifs.

3.2.1 Choix de l'espace de représentation (contexte)

Pour surveiller un système, il est nécessaire de disposer d'informations provenant de celui-ci, informations généralement délivrées par des capteurs. Néanmoins, les mesures des capteurs ne permettent pas toujours de définir les caractéristiques représentatives des différents états de fonctionnement et un prétraitement est souvent nécessaire. Il faut être attentif à la technique employée pour ce traitement des données, surtout au niveau du temps de calcul, car il doit permettre le fonctionnement en ligne, et le retard dû au temps de calcul ne doit pas être trop pénalisant par rapport au temps de réponse nécessaire pour connaître l'état courant du processus surveillé. Dans tous les cas, le choix d'un prétraitement des données est lié au domaine d'application.

Si un prétraitement des données s'avère nécessaire pour extraire les informations, quelle que soit la technique utilisée, il est nécessaire de conserver l'interprétabilité des caractéristiques, très importantes tant pour la détection que pour le diagnostic. Une simplification trop importante de l'espace de représentation, lors de la phase de caractérisation du modèle de comportement du processus, peut être néfaste au moment de la reconnaissance. En effet, une simplification trop hâtive génère le risque de non-détection de modes de fonctionnement non présents dans l'étape d'apprentissage.

Il s'agit donc de définir à partir des données, prétraitées ou non, un nombre D de caractéristiques ou descripteurs pertinents vis-à-vis des différentes situations ou modes de fonctionnement connus. Il n'existe pas de règle; il faut trouver un compromis entre la dimension D du vecteur et la pertinence des caractéristiques. Très souvent, ce choix est guidé par l'expertise sur le procédé.

Une fois l'espace de description choisi, il faut mettre en place l'ensemble de données pour l'apprentissage. Cet ensemble est constitué à partir des observations historiques du procédé. Il se peut que les différents modes de fonctionnement ne se trouvent pas dans un seul ensemble de données et la nécessité d'établir un espace de description généra,l indépendant de la base d'apprentissage, s'impose. Dans le cas de la méthode LAMDA, l'opérateur expert doit définir :

- l'intervalle d'opération, c'est-à-dire les valeurs minimale et maximale pour la normalisation des descripteurs quantitatifs et,
- toutes les modalités pouvant apparaître dans chacun des descripteurs qualitatifs.

De cette façon, l'opérateur pourra établir plusieurs combinaisons de configuration pour chaque descripteur dans l'espace de représentation.

A titre d'exemple, la Figure 3.1 illustre trois intervalles différents pour représenter un descripteur quantitatif. Le graphe **b)** montre la normalisation utilisant directement la valeur maximale et la valeur minimale présente dans la base d'apprentissage. Le graphe **a)** montre la normalisation obtenue en ne considérant que le rang dans lequel le descripteur doit se maintenir pendant le mode de fonctionnement

normal. Dans ce cas, les oscillations sont agrandies et les perturbations amplifiées. Au contraire, le graphe **c)** représente le descripteur normalisé à partir de toute la plage des valeurs théoriquement possibles et les petites perturbations ne sont pas observables.

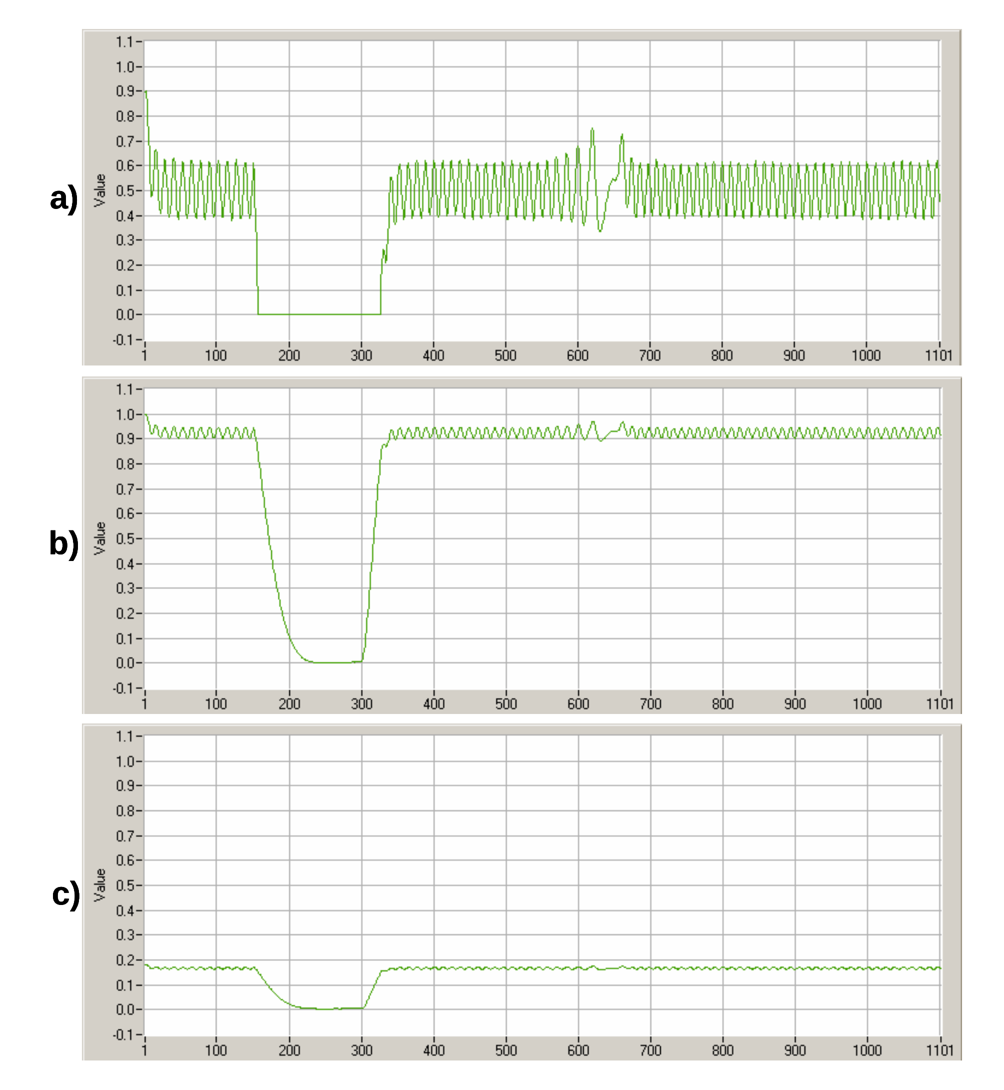


Figure 3.1. Choix de l'intervalle pour la normalisation d'un descripteur quantitatif

Le fait de pouvoir ajuster l'intervalle des descripteurs quantitatifs permet de définir jusqu'à quel point leurs variations peuvent être considérées. De même, le réglage de l'intervalle peut déterminer l'influence des descripteurs sur la caractérisation des classes.

3.2.2 Modes d'Apprentissage

Afin de déterminer la partition de l'espace de description la plus convenable pour caractériser les différents modes de fonctionnement, l'opérateur peut faire appel aux différents modes d'apprentissage en fonction de ses connaissances. En effet, quand seule la connaissance des experts est utilisée pour l'identification des différents états fonctionnels du processus, plusieurs situations peuvent surgir :

- l'expert n'est capable d'exprimer qu'une connaissance partielle du processus,
- il est conscient de l'existence de plusieurs états mais il ignore comment les identifier en utilisant les données historiques et celles disponibles en ligne,
- il n'a pas une idée claire des états à reconnaître.

Pour ces raisons la méthode de classification à utiliser doit fournir à l'expert plusieurs alternatives d'apprentissage pour identifier et caractériser les états fonctionnels.

Ainsi, la séparation en régions de l'ensemble des observations à utiliser pour l'apprentissage peut se réaliser selon un étiquetage des observations imposé par l'expert (apprentissage supervisé), ou de façon automatique (apprentissage non-supervisé). Une combinaison des deux types d'apprentissage peut être également envisagée.

3.2.2.1 Apprentissage supervisé - Imposé par l'expert

Lorsque l'opérateur connaît *a priori* les modes réels de fonctionnement contenus dans l'ensemble d'apprentissage, nous parlons d'un apprentissage *dirigé* ou *imposé* par l'expert. En effet, l'expert étiquette les observations qui d'après lui, représentent le mieux les différentes situations. De cette façon, les paramètres de chaque classe seront une combinaison des caractéristiques des observations imposées par l'expert.

3.2.2.2 Apprentissage non supervisé (auto-apprentissage)

Il n'est pas toujours possible de disposer des connaissances *a priori* sur les caractéristiques des modes de fonctionnement et l'expert ne peut pas affecter chaque observation à une situation. L'objectif est de proposer une structure de classes, dans un ensemble d'observations, en trouvant des ressemblances entre ces observations. Autrement dit, l'auto-apprentissage consiste en la **découverte**, à partir des informations contenues dans un ensemble de données, de classes ou groupes de classes caractérisant les différents modes de fonctionnement.

De cette façon, la méthode de classification propose une première partition à l'expert qui, ensuite, doit valider ces résultats en prenant en compte sa connaissance ainsi que les lois physiques régissant le comportement du processus.

3.2.2.3 Apprentissage supervisé - Actif

Nous pouvons constater qu'entre l'apprentissage dirigé (imposé) et l'auto-apprentissage il existe une troisième situation d'apprentissage pour laquelle une information partielle sur les données d'apprentissage est disponible. En fait, l'expert n'a pas besoin d'étiqueter toutes les observations de l'ensemble d'apprentissage, il peut laisser des éléments sans classe, soit parce qu'il considère que certains ne sont pas assez représentatifs, soit parce qu'il ne possède pas les connaissances pour déterminer si ces éléments correspondent à un type de comportement.

A ce moment là, l'algorithme d'apprentissage peut effectuer une première classification pour obtenir les paramètres de classes. Ensuite, les observations qui n'ont

pas été attribuées à une classe seront utilisées, soit pour faire évoluer les classes existantes, soit pour en créer de nouvelles.

Cette même procédure peut être réalisée d'une autre façon : dans un premier temps, un auto-apprentissage donnant lieu à une classification est réalisé. Dans un deuxième temps, à l'aide d'un nouveau jeu de données, un autre apprentissage est effectué, celui-ci autorisant la modification et/ou la création des classes obtenues précédemment.

Avec LAMDA, il est possible de mettre en place aussi bien l'auto-apprentissage que l'apprentissage dirigé (supervisé) ou la combinaison des deux (Apprentissage supervisé actif).

3.2.3 Réglage des paramètres du classificateur

Comme nous l'avons déjà indiqué, quand nous faisons appel aux algorithmes de classification pour déterminer les différents modes de fonctionnement à partir de données, en plus du choix du type d'apprentissage, il faut aussi choisir et ajuster certains paramètres de façon à obtenir une partition satisfaisante. Le réglage de ces paramètres est fait en fonction des caractéristiques de l'ensemble de données mais, aussi, en fonction des performances pendant la reconnaissance des nouvelles données.

En ce qui concerne la méthode LAMDA, il y a essentiellement trois paramètres à régler : la fonction d'adéquation marginale pour les descripteurs quantitatifs, les connectifs mixtes d'association pour l'agrégation des contributions de chaque descripteur et l'indice d'exigence, pour rendre plus ou moins stricte l'attribution d'un individu à une classe.

3.2.3.1 Fonctions d'adéquation marginale

Le choix de la fonction d'adéquation marginale est propre à la topologie ou nature de chaque descripteur ou de l'ensemble des descripteurs de l'espace de représentation.

Selon la topologie des descripteurs de type quantitatif, nous pouvons différencier trois cas :

- les valeurs des données pour un ou plusieurs descripteurs suivent une distribution de type normale autour d'un centre (ou prototype); la fonction de type *gaussien* (voir l'équation (17)) peut être sélectionnée,
- les valeurs des descripteurs se trouvent dans l'intervalle [X_{MIN}, X_{MAX}] et distribuées vers une des extrémités de cet intervalle, alors la fonction *Binomiale floue* (équation(14)) est la plus adéquate,
- si les valeurs des descripteurs ne suivent pas une distribution de type normal mais se trouvent autour d'un centre la fonction *Binomiale centrée* (équation (15)) est une alternative.

De cette façon, si l'opérateur possède des connaissances sur la distribution de chaque descripteur, il peut choisir la fonction du DAM (Degré d'Adéquation Marginal) la

plus appropriée; si ce n'est pas le cas ou s'il n'est pas satisfait de la classification obtenue, il peut choisir une autre fonction.

3.2.3.2 Opérateurs d'agrégation (connectifs) et indice d'exigence

Comme nous l'avons mentionné précédemment, il existe différentes possibilités pour agréger les contributions marginales (DAMs) d'un individu à une classe. L'expert a le choix parmi les familles de connectifs les plus connues : le *minimum* (T-norme) et le *maximum* (T-conorme) de la théorie de la Logique Floue de Zadeh [ZAD78] et celles du *produit* et *somme probabiliste* qui présuppose un raisonnement de type statistique avec une indépendance entre descripteurs. D'autres familles de connectifs pourraient être expérimentées; cependant dans ce travail nous nous limiterons à ces deux t-normes et t-conormes.

Tableau 3.1. Familles d'opérateurs d'agrégation

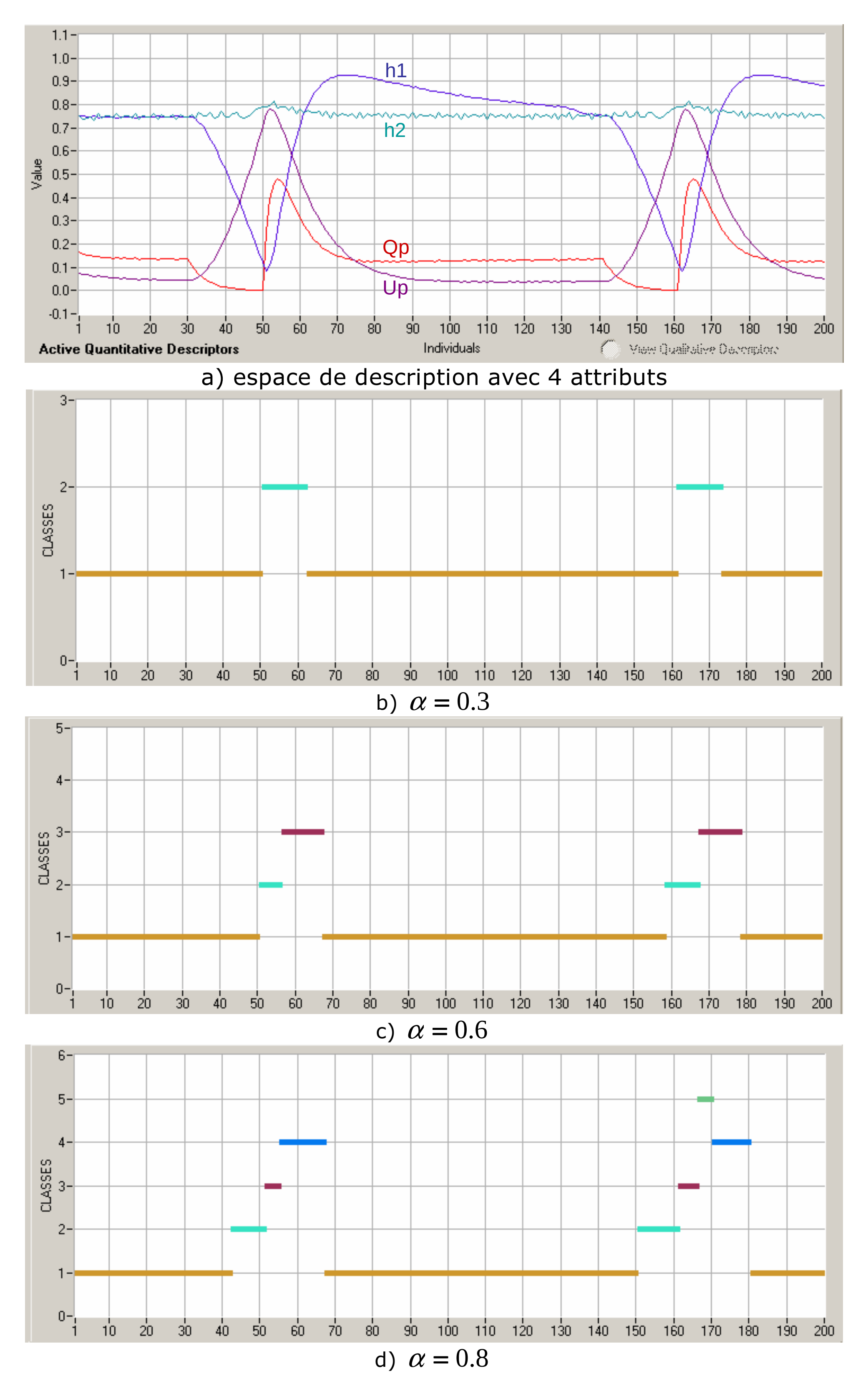
CONNECTIF	T – Norme (Intersection) $\boldsymbol{T}(\boldsymbol{x},\boldsymbol{y})$	T – Conorme (union) $\boldsymbol{S}(\boldsymbol{x},\boldsymbol{y})$
Produit / Somme probabiliste	$\boldsymbol{x}\cdot\boldsymbol{y}$	$\boldsymbol{x}+\boldsymbol{y}-\boldsymbol{x}\cdot\boldsymbol{y}$
Minimum / Maximum	$\min(\boldsymbol{x},\boldsymbol{y})$	$\max(\boldsymbol{x},\boldsymbol{y})$

L'algorithme d'auto-apprentissage de LAMDA étant séquentiel, avec des connectifs différents, différentes partitions seront générées.

De plus, comme nous l'avons déjà mentionné dans le chapitre 2 (§2.3.5), il est possible de régler l'exigence du classificateur à l'aide de l'*indice d'exigence* α. Ainsi, le classificateur le plus exigeant accepte un individu dans une classe uniquement si tous ses attributs présentent un DAM élevé à cette classe; ce qui correspond à avoir $\alpha = 1$ (T-Norme (intersection)). De la même manière, le classificateur le moins exigeant accepte un individu dans une classe si au moins l'un de ses attributs possède une appartenance marginale élevée à cette classe; dans ce cas il s'agit de la T-Conorme ($\alpha = 0$). Donc, à partir du *connectif mixte*, l'expert peut avoir une série de partitions plus ou moins strictes de l'espace de description.

La Figure 3.2 montre un exemple des résultats avec un espace de représentation à 4 descripteurs. Les données représentent un mode de fonctionnement normal et une situation de défaillance avec retour en fonctionnement normal (Figure 3.2a); la défaillance est présente deux fois. Les différentes partitions illustrées ont été obtenues en faisant de l'auto-apprentissage utilisant la fonction *binomiale floue* et, comme opérateurs d'agrégation, le Minimum (T-Norme). Dans la Figure 3.2b), une exigence très faible ($\alpha = 0.3$) est choisie et le classificateur n'a identifié que 2 classes. Au fur et à mesure que l'exigence est incrémentée, le nombre de classes augmente (Figure 3.2c, d) jusqu'à obtenir une partition satisfaisante. Dans notre exemple, la classification qui a donné la partition, considérée la plus significative par l'expert, a été obtenue avec une exigence de 0.8, pour laquelle 5 classes ont été identifiées.

Le fait de pouvoir contrôler l'exigence de l'algorithme de classification permet à l'expert de trouver un compromis entre une partition suffisamment représentative des modes de fonctionnement et une partition capable de faire de la généralisation selon les besoins.



a) espace de description avec 4 attributs

b) $\alpha = 0.3$

c) $\alpha = 0.6$

d) $\alpha = 0.8$

Figure 3.2. Exemples de classification pour différentes valeurs d'*exigence* α.

3.3 Analyse et interprétation des résultats

Il s'agit de fournir à l'expert des outils visuels avec des informations utiles relatives à la classification réalisée. L'objectif est de donner à l'expert des moyens pour mieux exploiter les résultats de la classification lors de l'interprétation et la validation des classes.

3.3.1 Reconnaissance de l'ensemble d'apprentissage

Nous considérons que pendant la phase d'apprentissage il existe une interaction constante entre la création et la reconnaissance des classes. Après qu'une classification ait été définie, il s'agit de traiter l'ensemble d'apprentissage et d'assigner aux classes existantes les individus de l'ensemble d'apprentissage ou d'un ensemble dans lequel l'expert connaît en partie les attributions correctes. La reconnaissance doit disposer du choix de "rejet", c'est-à-dire qu'il faut pouvoir distinguer les observations qui ne correspondent à aucune des classes existantes, soit parce qu'elles se trouvent dans une zone de l'espace de représentation "trop éloigné" des classes, soit parce qu'elles se trouvent "près" d'une frontière entre deux ou plusieurs classes, ce qui correspond à une situation d'*ambiguité*. En effet, des nouvelles situations de défaillances ou des dégradations peuvent se présenter au cours de la vie d'un procédé et il serait dangereux d'affecter systématiquement une nouvelle observation à une situation existante (connue). Dans la littérature, une première réponse au problème d'ambiguïté a été introduite par Chow [CHO70]. Plus récemment Dubuisson et Masson [DUB93] ont établi une distinction entre le *rejet de distance* et le *rejet d'ambiguïté*. La Figure 3.3 illustre, pour deux classes de type gaussien, les zones de rejet d'ambiguïté et de rejet de distance.

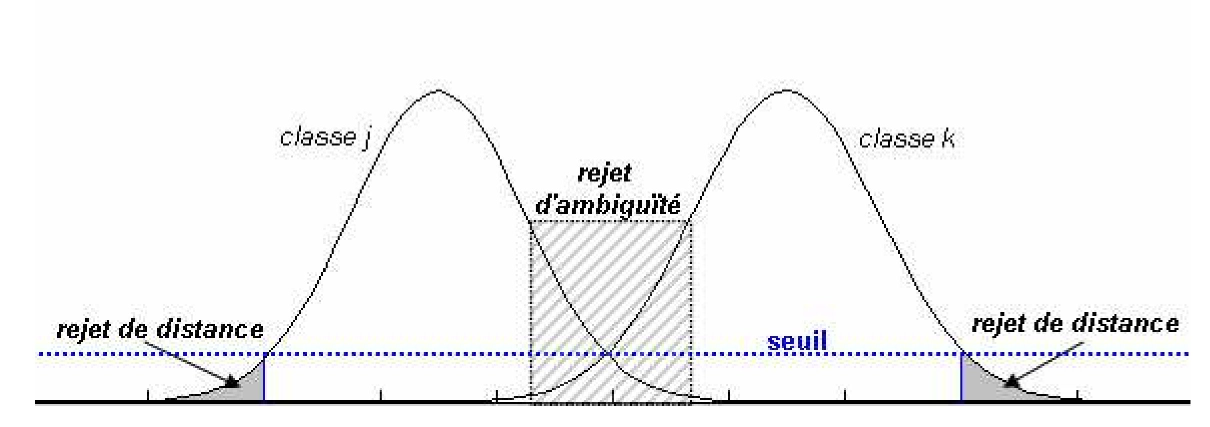
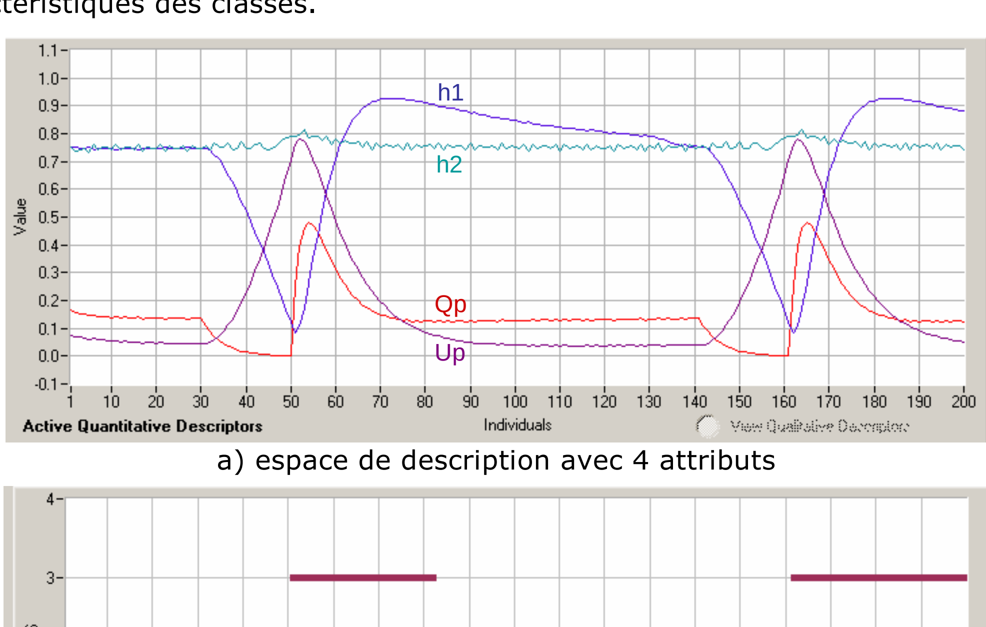


Figure 3.3. Zones de rejet d'ambiguïté et de distance.

En ce qui concerne la méthode LAMDA, la notion de rejet est considérée en prenant en compte le seuil de minimum d'appartenance (DAG_{NIC}) de la classe *Non-Informative (NIC)* de telle façon à ne pas attribuer des observations avec une trop faible appartenance. Le cas de l'ambiguïté ne donne pas lieu à une zone de non classement, nous avons préféré adopter le critère de décision du "premier trouvé", certes arbitraire, mais la fréquence de cet événement s'avère faible.

Dans le cas où un apprentissage dirigé (imposé) a été effectué, il faut d'une part que l'expert valide les caractéristiques des classes pour la représentation des modes de fonctionnement, d'autre part, qu'il connaisse les performances du classificateur, c'est-à-dire savoir combien d'individus n'ont pas été reconnus, combien ont été correctement classés et lesquels ont été mal classés (voir Figure 3.4). Ces informations permettent à l'expert de :

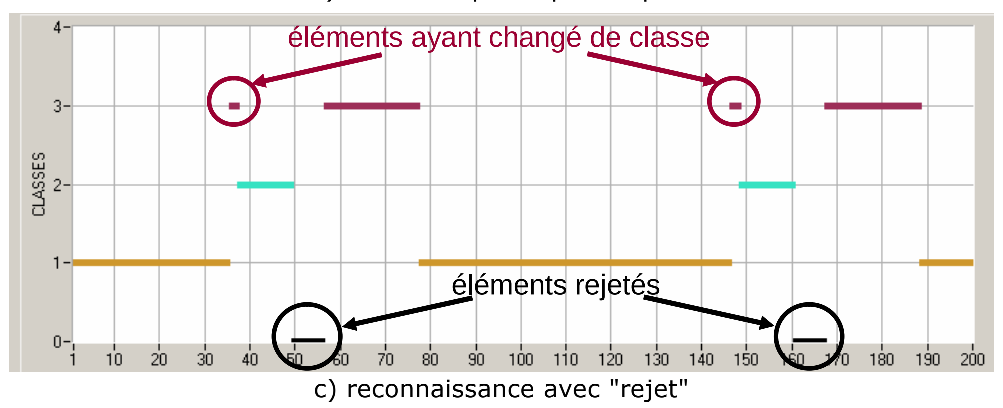
- modifier les paramètres du classificateur, pour obtenir une meilleure généralisation ou plus de sélectivité,
- vérifier si les individus mal reconnus ont, peut être, été mal étiquetés car leurs caractéristiques sont effectivement plus proches de la classe à laquelle ils ont été attribués lors de la reconnaissance,
- vérifier également les caractéristiques des individus non reconnus par rapport aux caractéristiques des classes.



a) espace de description avec 4 attributs



b) classes imposés par l'expert



c) reconnaissance avec "rejet"

Figure 3.4. Reconnaissance après une classification avec apprentissage dirigé.

Dans le cas d'un auto-apprentissage, la reconnaissance de l'ensemble d'observations utilisées pour la création des classes permet à l'expert de valider si les observations ont été affectées dans la même classe que celle de la phase d'apprentissage. L'affectation d'une observation à une classe différente peut se produire à cause de la mise à jour et de la création de classes réalisées de manière séquentielle. En effet, une observation peut avoir une appartenance plus grande à une classe créée que celle à laquelle elle a été associée. Il est donc souhaitable de procéder hors ligne à un auto-apprentissage qui traite plusieurs fois l'ensemble de données, jusqu'à ce qu'une proportion, définie comme acceptable, d'éléments ait changé de classe entre deux itérations.

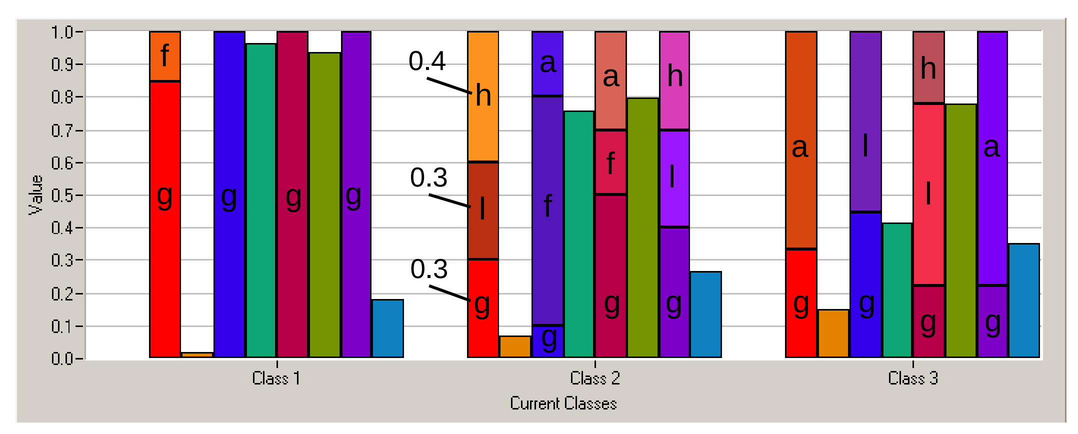
3.3.2 Analyse du pouvoir d'extrapolation

Pour tester les performances du classificateur et valider ses capacités de généralisation, l'utilisation d'autres ensembles de données dans le même espace de description est souhaitable. De la même façon que pour la reconnaissance de l'ensemble d'apprentissage, la reconnaissance des individus de test doit permettre à l'expert d'accepter des individus non reconnus.

Il est souhaitable que parmi les observations de test ne se trouvent pas des situations non apprises pendant la phase d'apprentissage, cependant dans l'éventualité d'une telle présence, il vaut mieux permettre le "rejet" des observations qui n'ont pas une appartenance aux classes supérieure au DAG (Degré d'Adéquation Global) de la NIC (Non_Informative Class). Les observations qui ont été assignées à la NIC peuvent être utilisées par la suite pour la création des nouvelles classes ou, pour faire évoluer les classes déjà existantes en mode *apprentissage supervisé actif*.

3.3.3 Interprétation du profil des classes

Le ***profil*** de chaque classe est représenté par la moyenne calculée des valeurs normalisées des descripteurs quantitatifs et pour les qualitatifs, par la fréquence d'apparition de chaque modalité. Ceci montre les caractéristiques de chaque classe selon l'espace de description, calculées d'après le type d'algorithme employé pour l'apprentissage. La Figure 3.5 illustre les caractéristiques des classes obtenues après un apprentissage dirigé. Il y a trois classes, chaque classe est caractérisée par 8 descripteurs, 4 quantitatifs et 4 qualitatifs. Les descripteurs quantitatifs correspondent au paramètre ρ_{kj} et les descripteurs qualitatifs représentent les épisodes de chaque variable, où chaque modalité correspond à un type d'épisode (§1.3.3.1). Dans la Figure 3.5 les descripteurs du profil des classes apparaissent dans l'ordre suivant : Up_Ep (qualitatif), Up_Val (quantitatif), h2_Ep (qualitatif), h2_Val (quantitatif), h1_Ep (qualitatif), h1_Val (quantitatif), Qp_Ep (qualitatif) et Qp_Val (quantitatif).



Episodes : g = γ, f = ϕ, a = α h = η, l = λ

Figure 3.5. Profil des classes obtenues à partir d'un apprentissage dirigé

À partir du profil de classes l'expert peut déterminer quels sont les descripteurs les plus discriminants dans l'espace de représentation, c'est-à-dire ceux qui ont le plus d'influence dans la caractérisation des classes. Pour ce faire, il est possible de calculer l'*entropie* de chaque descripteur par rapport à chaque classe. Ainsi le descripteur avec la plus petite entropie sera le plus discriminant, et celui avec la plus grande entropie est celui qui apporte le moins dans la caractérisation des classes.

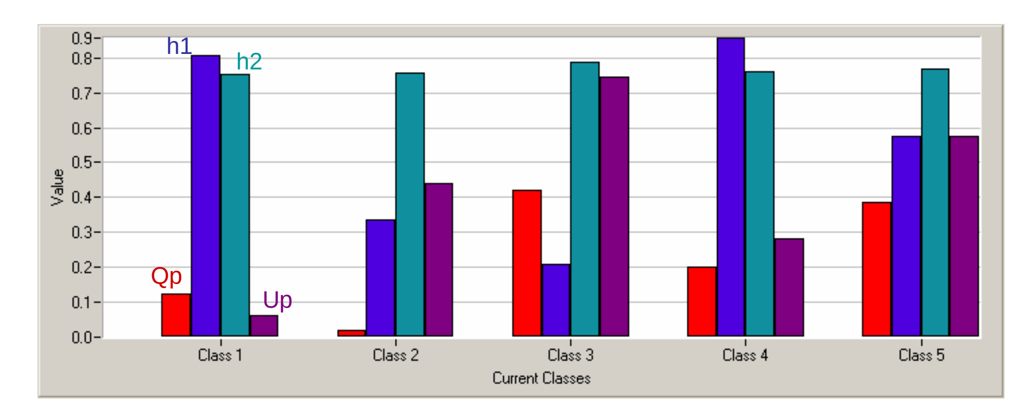


Figure 3.6. Profil des classes avec descripteurs quantitatifs

Pour l'exemple de la Figure 3.5, le descripteur "*Qp*" est le plus discriminant et le moins discriminant est "*h2*" puisque ses valeurs pour chaque classe sont peu différentes.

Ces informations permettent à l'expert d'identifier et de caractériser les classes significatives et celles qui pourraient représenter un même état fonctionnel.

3.3.4 Visualisation et interprétation des appartenances

Une des informations utiles relative à la classification réalisée, est la ***matrice d'appartenances*** des individus à chaque classe. Dans cette matrice, (29) les lignes correspondent aux individus et les colonnes aux classes existantes. $\boldsymbol{DAG_{nk}}$ est alors le degré d'adéquation global de l'individu *n* à la classe *k*.

$$\begin{bmatrix} DAG_{11} & \cdots & DAG_{1K} \\ \vdots & DAG_{nk} & \vdots \\ DAG_{N1} & \cdots & DAG_{NK} \end{bmatrix} \quad (29)$$

3.3.4.1 Evolution des degrés d'adéquation globale (DAG)

Étant donné que l'ordre d'apparition des éléments correspond aux instants de mesure des variables du processus, la représentation graphique de la **Matrice d'Appartenance (MA)**, en prenant l'ordre des lignes comme axe temporel, illustrée par la Figure 3.7, permet de visualiser deux aspects importants de la classification par rapport au degré d'adéquation global (DAG) de chaque individu aux classes :

- **la distinction entre classes** : lorsque le DAG des individus appartenant à une classe est suffisamment éloigné du DAG de la NIC ainsi que de ceux des autres classes, l'opérateur peut constater que la classe a été bien distinguée. Cette information est pertinente au moment d'identifier une situation significative sur le comportement du procédé (voir DAG pour la classe *Cl 2* de la Figure 3.7),
- **la proximité entre classes** : en revanche, quand les DAGs d'une ou plusieurs classes ont des évolutions similaires l'expert peut considérer la possibilité de regrouper plusieurs classes pour caractériser un même mode de fonctionnement. Ceci est illustré dans la Figure 3.7 par la proximité des DAGs pour la classe C_1 et la classe C_4. Un indicateur de la proximité entre classes est donné par le calcul de la connectivité entre classes (§3.3.4.2). Une analyse des corrélations entre les courbes représentatives de l'évolution des DAGs pourrait apporter une aide dans ce sens.

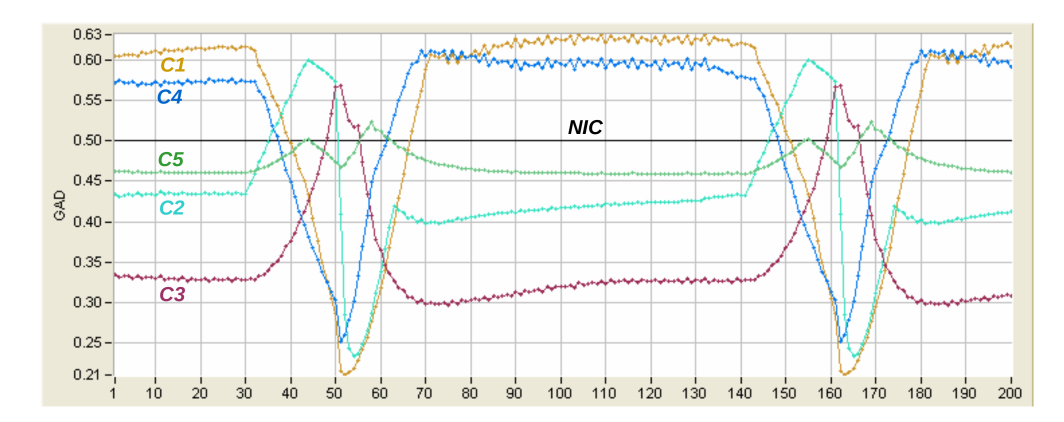


Figure 3.7. Visualisation des DAGs pour chaque classe

3.3.4.2 Matrice de connectivité

La "*connexion*" entre classes est définie par le nombre d'individus qui sont partagés par deux classes et qui ont un *DAG* supérieur à la valeur d'un seuil fixé par l'expert que nous appèlerons *α-cut* [AGUA98] selon l'usage en théorie des ensembles flous.

CLASSES CONECTIVITY MATRIX

	Class 1	Class 2	Class 3	Class 4	Class 5
Class 1	186	0	0	0	0
Class 2	0	39	0	0	0
Class 3	0	0	15	0	0
Class 4	0	0	0	54	0
Class 5	0	0	0	0	18

a) *α-cut = 1.0*

CLASSES CONECTIVITY MATRIX

	Class 1	Class 2	Class 3	Class 4	Class 5
Class 1	219	3	0	48	0
Class 2	3	39	3	0	0
Class 3	0	3	18	0	0
Class 4	48	0	0	75	6
Class 5	0	0	0	6	21

b) *α-cut = 0.97*

CLASSES CONECTIVITY MATRIX

	Class 1	Class 2	Class 3	Class 4	Class 5
Class 1	312	312	312	312	312
Class 2	312	312	312	312	312
Class 3	312	312	312	312	312
Class 4	312	312	312	312	312
Class 5	312	312	312	312	312

c) *α-cut = 0.0*

Figure 3.8. Connectivité entre classes pour l'exemple de la **Figure 3.7**

L'opérateur peut alors regrouper plusieurs classes car il considère que les caractéristiques des individus partagés conviennent aussi bien à une classe qu'à une autre, c'est-à-dire que leurs DAGs sont très proches et supérieur à l'*α-cut*. Une analyse pour plusieurs valeurs de α est réalisée pour déterminer la meilleure union entre classes. A noter que, pour *α-cut=1,* toutes les classes sont disjointes (Figure 3.8a)) et, pour *α-cut=0* (Figure 3.8c)), elles sont toutes connectées.

3.3.5 Affectation des classes en états fonctionnels

Lors d'un auto-apprentissage, l'interprétation des classes en terme de modes de fonctionnement ou états fonctionnels est indispensable. En effet, certaines classes obtenues présentent parfois des caractéristiques très proches et l'expert peut décider d'affecter ces classes à un seul état. Pour réaliser l'agrégation des classes dans un seul état significatif, l'opérateur peut alors s'appuyer sur la représentation graphique du profil de classes et de la similitude des valeurs des profils (§ 3.3.3), ainsi que sur la matrice de connectivité (§ 3.3.4). Dans le cas de l'exemple de la Figure 3.2a) et à partir des interprétations issues de la Figure 3.6 et de la Figure 3.7, l'opérateur peut éventuellement regrouper les classes C_1 et C_5 dans un même état fonctionnel.

3.4 Reconnaissance des classes en ligne

Une fois que le modèle de référence du comportement connu a été élaboré, l'étape suivante consiste à déterminer, à chaque instant, dans quel état fonctionnel se trouve le procédé lorsqu'une nouvelle observation est présentée. Il faut rappeler que tous les modes de fonctionnement ne sont pas forcement identifiés lors de l'étape d'apprentissage et, donc, le modèle de référence (comportement) n'est pas exhaustif. Ceci est dû, par exemple, au fait que pour certaines situations le recueil des données n'a pas été possible ou qu'il existe des situations dont l'occurrence est peu fréquente.

3.4.1 Suivi des états attendus

La reconnaissance des situations attendues établies lors de l'apprentissage à partir des nouvelles observations est normalement passive, c'est-à-dire que les classes ne sont pas modifiées lorsqu'un nouvel élément est pris en compte. Cependant, il est aussi possible de prévoir l'éventualité de faire évoluer les classes mais, à ce moment là, pour réaliser la modification d'une classe l'algorithme LAMDA, a besoin d'avoir défini à l'avance le nombre ***N*** d'éléments contenus dans la classe au moment de sa mise à jour. Ce paramètre peut rester fixe ou être incrémenté chaque fois qu'un nouvel élément est affecté. Dans le cas où la valeur de *N* est fixée, ceci correspond à un oubli exponentiel, où, en pratique, seulement les *N* derniers éléments vont contribuer à définir les caractéristiques de la classe et la classe va suivre la trajectoire de ces éléments. Le choix de cette valeur est très important car si une valeur trop petite est choisie, les caractéristiques de la classe seront trop similaires aux attributs des derniers éléments classés. Dans le cas où *N* est incrémenté à chaque fois qu'un individu est affecté, il faut déterminer la valeur initiale de *N.*

Dans le cadre de notre travail, nous avons préféré ne pas modifier les classes lors de la reconnaissance en ligne car, en absence d'information sur l'éventuelle évolution des classes, nous considérons très risqué le fait de les modifier au fur et à mesure de l'affectation des individus. En effet, dans ce cas, il sera possible qu'une situation définie dans le cadre du fonctionnement normal évolue vers une situation anormale (de défaillance) sans que l'opérateur soit capable de reconnaître en ligne le fait qu'il y a eu un problème. Lorsqu'une observation commence à "s'éloigner" de tous les modes de fonctionnement connus, son DAG maximal se rapproche de plus en plus du seuil minimal (DAG_{NIC}) jusqu'à ne plus être affecté à une classe existante, ce qui conduit à la détection d'une déviation de comportement du processus surveillé.

Pour savoir quel est l'état courant du procédé surveillé, les différentes appartenances (DAG) de l'élément correspondant à l'instant d'observation sont calculées et l'élément est affecté à la classe pour laquelle son DAG est maximum.

3.4.2 Détection de déviations– éléments non reconnus ou mal reconnus

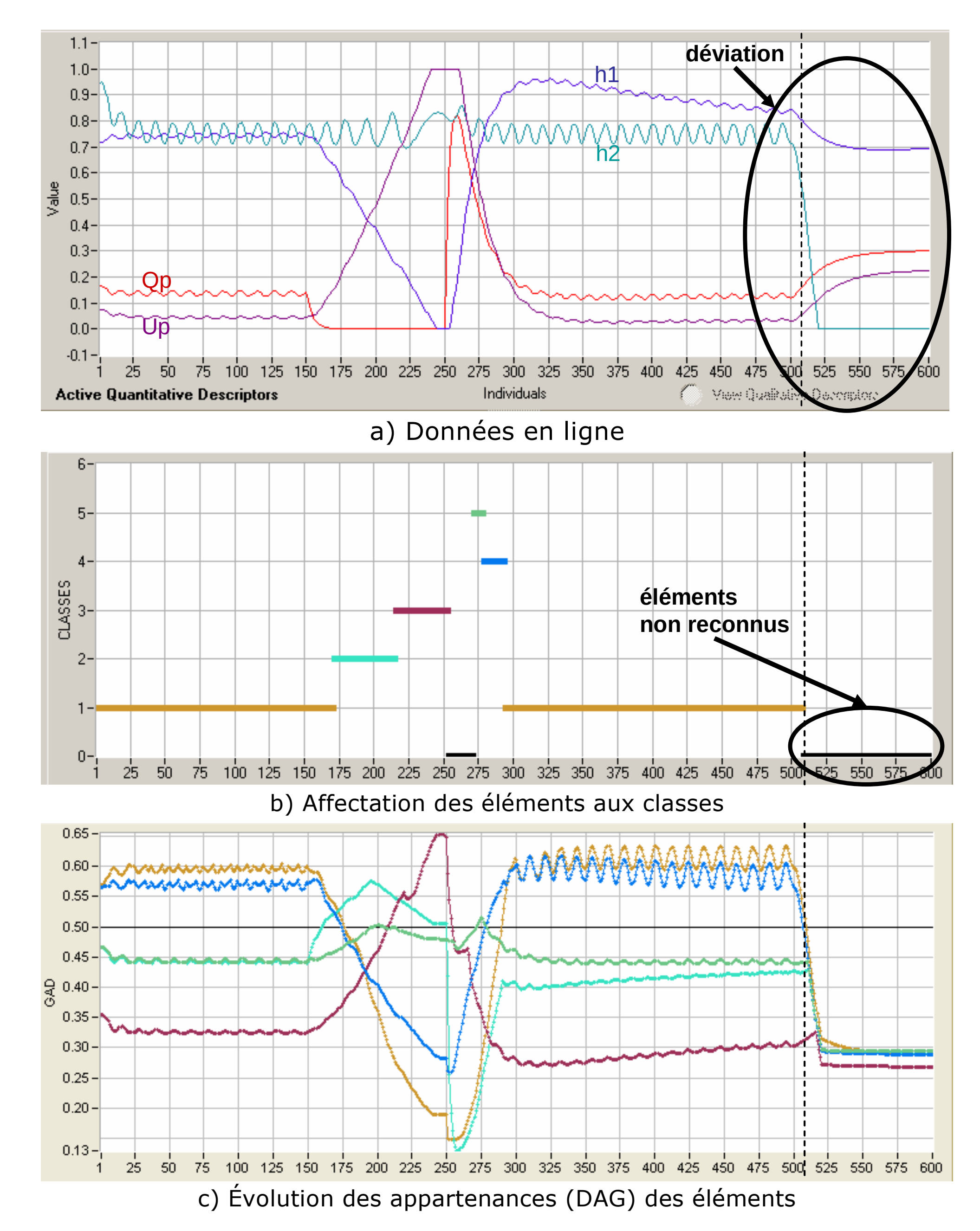
Du fait qu'une reconnaissance passive en ligne est réalisée, le suivi ne peut que confirmer une situation existante (attendue) dans le modèle de référence ou bien rejeter un nouvel élément car il est trop différent et ne peut être affecté à aucune classe. Il y a deux situations défavorables qui peuvent se produire :

- les nouveaux éléments sont reconnus dans une classe existante qui n'est pas celle qui leur correspondait : nous considérons qu'ils sont mal reconnus,
- les nouveaux éléments ne sont reconnus dans aucune classe.

Il se peut en effet qu'un élément soit mal reconnu à cause du bruit présent dans les mesures constituant les attributs. Donc, l'élément peut avoir des appartenances similaires entre deux ou plusieurs modes de fonctionnement et son affectation à l'un d'eux induit un risque de mauvais classement. Ce mauvais classement peut induire des fausses alarmes dans le système et, s'il y a un nombre significatif d'éléments mal classés, il peut être nécessaire de remettre en question la façon selon laquelle le modèle de référence a été conçu pendant la phase d'apprentissage.

Un ***symptôme*** est identifié par la déviation du comportement attendu pendant la phase de reconnaissance en ligne. Ces déviations sont détectées lorsqu'un nombre suffisant d'éléments sont affectés à la *NIC*, c'est-à-dire que leur appartenance aux classes existantes ne dépassent pas le seuil minimum d'adéquation et donc qu'ils ne correspondent à aucun comportement prévu.

Dans la Figure 3.9, un exemple de la phase de reconnaissance en ligne est illustré pour une période de 600 secondes (un individu par seconde). Pendant les 500 premiers éléments, les différentes classes (comportements) connues ont été correctement identifiées. À partir de l'élément 520, une déviation a été détectée par l'affectation des éléments à la *NIC*. Comme le montre la Figure 3.9c), les appartenances des éléments pour toutes les classes existantes ont des valeurs inférieures au seuil de minimum d'adéquation (dans cet exemple $DAG_{NIC} = 0.5$). Même si l'opérateur est alerté par le rejet de certains éléments, aucune information sur les causes ou la nature du symptôme n'est disponible. Il faut alors lancer une procédure d'apprentissage pour caractériser le symptôme qui vient d'être mis en évidence.



a) Données en ligne

b) Affectation des éléments aux classes

c) Évolution des appartenances (DAG) des éléments

Figure 3.9. Reconnaissance en ligne - suivi et détection des déviations

3.4.3 Caractérisation des déviations détectées

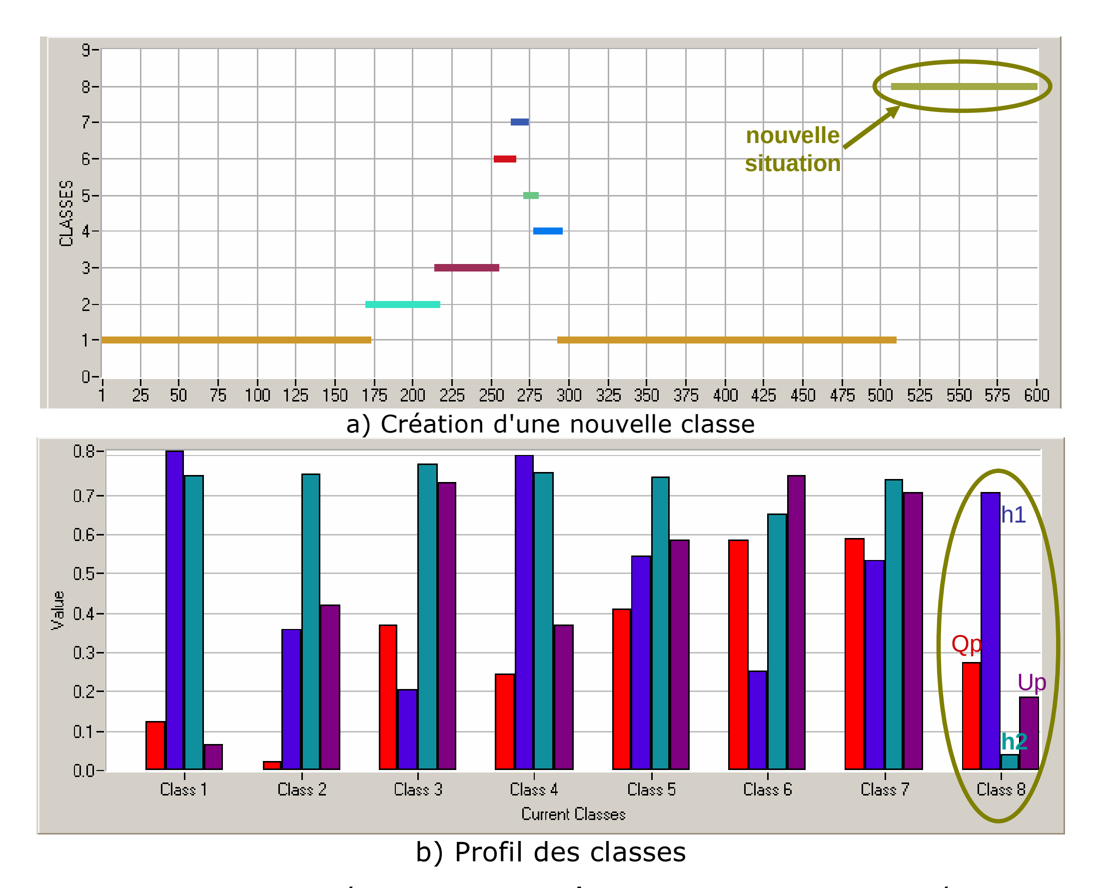
Dès qu'un nombre significatif d'éléments consécutifs sont affectés à la classe non-informative (NIC), c'est-à-dire que leur appartenance aux classes C_k ne dépassent pas le seuil minimum d'adéquation, la caractérisation de ces éléments peut être réalisée en lançant une nouvelle étape d'apprentissage. Différents types d'apprentissage sont envisageables : l'*apprentissage hors ligne* réalisé de manière périodique et l'*apprentissage en ligne* réalisé en continu.

Au cours de l’apprentissage en ligne, toutes les observations présentées sont utilisées pour l'adaptation des classes ou la création de nouvelles. En effet, l'apprentissage en ligne consiste en un auto-apprentissage où, à partir d'un nombre

d'éléments non reconnus, l'algorithme de classification procède à la création d'une nouvelle classe, laquelle doit être interprétée ensuite par l'expert comme représentative ou non d'un nouveau mode de fonctionnement.

Dans le cadre de notre travail, nous nous sommes intéressée à la caractérisation des déviations à partir d'un apprentissage hors ligne. A partir des éléments non reconnus en ligne, l'expert peut lancer un nouvel apprentissage, pour créer des nouvelles classes et ainsi caractériser les déviations détectées par la non reconnaissance.

Le nouvel apprentissage prend en compte le classificateur construit auparavant et l'expert peut choisir soit de créer des nouvelles classes en gardant les caractéristiques des classes existantes inchangées, soit de créer des nouvelles classes et d'ajuster aussi les classes existantes. Cette deuxième procédure présente les dangers que nous avons évoqués lors de la modification des classes en ligne. Pour le cas de l'exemple de la Figure 3.9a), un apprentissage supervisé actif, avec la possibilité de création de classes, a été réalisé. Une nouvelle classe a été générée à partir des éléments rejetés, sans modification des paramètres des classes existantes. La Figure 3.10 illustre les résultats : la classe 8 a été créée ayant comme profil celui montré dans la Figure 3.10b). Nous pouvons constater qu'aucune des classes obtenues auparavant ne présentait un profil proche de celui de la nouvelle classe.



a) Création d'une nouvelle classe

b) Profil des classes

Figure 3.10. Caractérisation du symptôme - apprentissage supervisé actif

A partir des résultats obtenus de l'apprentissage supervisé actif, et avec une analyse du profil de la nouvelle classe, c'est-à-dire de ses caractéristiques, l'opérateur

peut interpréter la déviation détectée auparavant comme une situation normale mais non présente lors de la phase d'apprentissage, ou bien comme une défaillance. S'il s'agit d'une défaillance, alors les informations du profil de la classe peuvent être utilisées pour éventuellement identifier les causes (ou l'origine) de la défaillance.

Nous rappelons que notre objectif est d'aider l'expert dans son diagnostic et non de générer automatiquement un résultat de diagnostic.

Dans l'exemple de cette nouvelle situation (Figure 3.10), l'expert a identifié qu'il s'agit d'une situation de défaillance et que, selon les valeurs des descripteurs, le problème se trouve au niveau du descripteur "h2" qui est anormalement petit.

Une fois que la nouvelle situation a été identifiée, elle sera ajoutée au modèle de référence pour le mettre à jour, afin de le compléter et de continuer le suivi des états attendus.

3.5 L'outil SALSA

Pour mettre en oeuvre le système de surveillance à partir des données et en utilisant la méthode de classification LAMDA, il est nécessaire d'établir une interaction homme-machine. Cette interaction est faite au travers d'une visualisation des résultats obtenus par la classification, de manière à ce que l'expert puisse valider ou non la partition proposée. Brachman et Anand [BRA96] insistent sur l'importance d'une interface claire et efficace entre la procédure de découverte (création des classes dans le cas d'auto-apprentissage) et la validation par l'opérateur humain comme l'une des parties clés de toute méthodologie de *Data Mining.*

Ainsi, à partir des considérations décrites précédemment, nous avons développé un outil qui permet de mettre en œuvre la stratégie proposée dans le chapitre 2. Cet outil offre une interface modulaire, flexible et convivial qui intègre les données issues du procédé, l'algorithme de classification LAMDA et les connaissances de l'opérateur expert afin de l'assister pendant l'élaboration du modèle de comportement du système à surveiller. Par ailleurs, il apporte aussi une aide au moment du suivi et lors de la détection des déviations.

Nous avons appelé cet outil SALSA (*Situation Assessment using LAMDA claSsification Algorithm)* et son utilisation comprend deux étapes :

- Une étape hors ligne pour l'élaboration du modèle de référence. Pour ce faire, nous avons adapté la méthodologie de classification LAMDA de manière à ce que l'expert puisse intervenir de façon dynamique pendant la création et la reconnaissance de classes. La Figure 3.11 montre la structure de l'outil SALSA pour la phase d'apprentissage hors ligne. A partir des données historiques, l'opérateur modifie les paramètres de configuration jusqu'à obtenir un ensemble de classes représentatives du comportement du procédé. Quand la classification obtenue est acceptable, l'opérateur peut réaliser l'affectation des classes aux états fonctionnels identifiés

pour obtenir ainsi un ***modèle de référence***. Cette association des classes vers des états significatifs est ensuite visualisée de façon graphique lors de l'analyse.

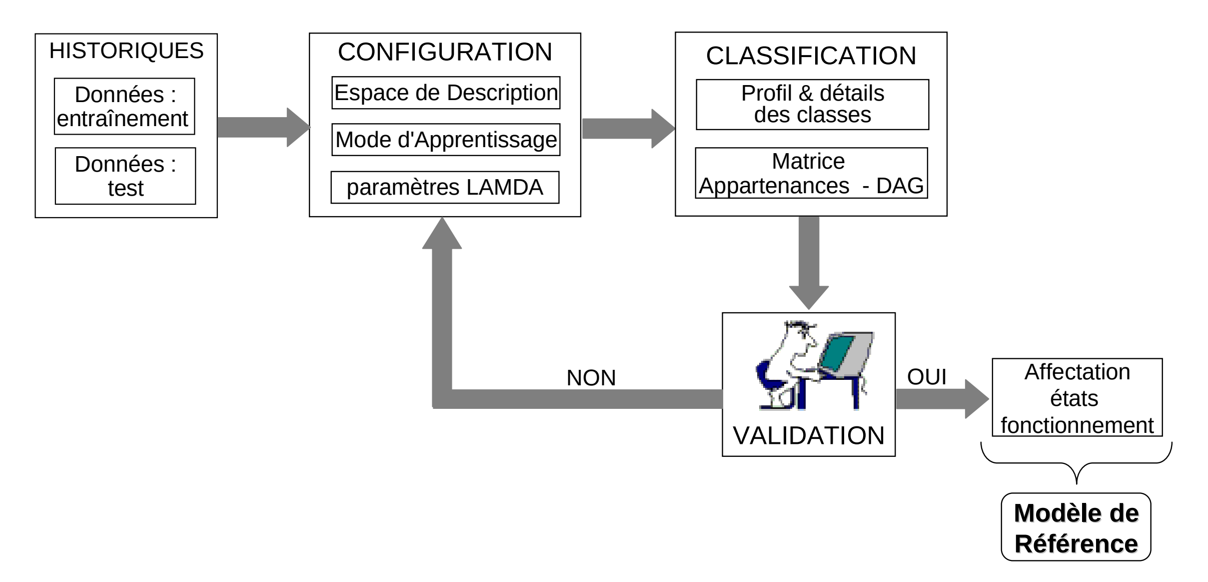


Figure 3.11. Structure SALSA hors ligne - Création du modèle de référence

- Une étape en ligne dans laquelle, à partir du modèle de référence élaboré hors ligne et des observations présentées en ligne, l'identification de l'état courant du procédé est réalisé (voir Figure 3.12). Cette étape effectue le suivi du procédé surveillé et alerte l'opérateur dès qu'une déviation est détectée, via des observations, correspondant à un état non reconnu (assigné à la NIC).

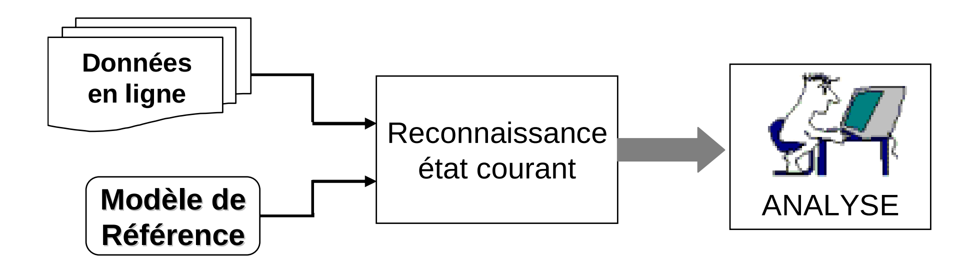


Figure 3.12. Structure SALSA en ligne - Identification des états fonctionnels

Dans l'annexe B nous décrivons de façon plus détaillée les modules d'apprentissage hors ligne et de reconnaissance en ligne de l'outil SALSA et leur utilisation.

3.6 Conclusion

Ce chapitre a été consacré à l'identification des différents états fonctionnels pour la surveillance des procédés. Nous avons décrit, dans un premier temps, les différents aspects à prendre en compte lors de l'élaboration du modèle de référence, c'est-à-dire pendant la phase d'apprentissage. Parmi ces aspects, nous trouvons :

- la définition de l'espace de description où l'opérateur doit choisir le type et la quantité d'attributs qui représentent le mieux le comportement du procédé,
- le mode d'apprentissage selon la connaissance *a priori* sur les situations représentées au sein de l'ensemble de données pour l'apprentissage,
- le réglage des paramètres de l'algorithme de classification choisi.

Bien que la connaissance experte soit toujours nécessaire pour la construction du modèle de référence, il faut apporter à l'opérateur les moyens pour faciliter l'analyse et l'interprétation des résultats obtenus lors de l'apprentissage. Ceci est réalisé au travers d'une interface homme-machine essentiellement graphique.

Une fois le modèle de référence construit, nous avons présenté l'étape d'identification de l'état courant du procédé à partir des données en ligne et nous avons détaillé les démarches à suivre lors de la détection d'une déviation. Dans ce cas, une caractérisation de la déviation (symptôme) est réalisée par un nouvel apprentissage hors ligne prenant en compte les éléments rejetés précédemment. A nouveau, c'est l'opérateur qui décide si la représentation obtenue de l'apprentissage est significative et doit être ajoutée au modèle de référence pour le compléter.

Différents exemples ont permis d'illustrer la mise en place de la stratégie proposée en utilisant la méthode LAMDA.

Finalement, nous avons présenté la structure de fonctionnement de l'outil SALSA. Cet outil a été conçu comme une interface home-machine pour assister l'opérateur humain dans le suivi du comportement du procédé ainsi que dans la détection et le diagnostic de défaillances.

Jusque là, nous avons identifié les différents états fonctionnels du procédé surveillé et détecté des déviations représentant d'éventuelles défaillances. Notre objectif, par la suite, est de construire un modèle des états fonctionnels représenté sous forme d'automate. De cette façon, nous aurons une représentation discrète d'un procédé continu et la possibilité de suivre les évolutions du procédé à partir des changements d'états. Ceci fait l'objet du chapitre suivant.

Chapitre 4

4. CONSTRUCTION D'UN AUTOMATE POUR LA SURVEILLANCE A L'AIDE DE LA CLASSIFICATION

4.1 Introduction

Le but de ce chapitre est de compléter l'identification de l'état fonctionnel dans lequel se trouve le processus en présentant à l'expert la séquence d'évolution des états d'opération du procédé.

Un des objectifs de notre travail est de mettre en place, à partir des résultats de la classification, un modèle des états fonctionnels et des transitions entre ces états. Ce modèle peut être considéré comme un modèle dynamique qui décrit les évolutions possibles des états fonctionnels du procédé.

Pour l'élaboration de ce modèle à états discrets, nous devons, d'une part, déterminer les états et, d'autre part, identifier les transitions entre états.

Pour illustrer la méthode de construction de l'automate, nous nous sommes appuyée sur un exemple dont les résultats seront détaillés dans le chapitre 5.

4.2 Rappels sur les machines à états finis

Nous avons introduit le concept d'état de fonctionnement pour représenter une situation dans laquelle se trouve un processus ou système dynamique. Ce concept fait appel à une structure dynamique connue sous le nom de ***machine à état fini*** ou ***automate***.

D'un point de vue formel, un automate est formé par trois ensembles dénombrables, finis, appelés alphabets U , X , Y et deux fonctions:

$$f : X \times U \rightarrow X \tag{30}$$

$$h : X \rightarrow Y \tag{31}$$

La fonction f est appelée *fonction de transition d'état* et la fonction h *fonction d'observation*. Nous avons introduit la possibilité de réunir plusieurs classes obtenues par apprentissage dirigé ou auto-apprentissage en un seul état fonctionnel; cette opération correspond à la fonction h de mesure. Les caractéristiques dynamiques utiles à la surveillance ne dépendent pas de cette fonction qui est uniquement une application passive. Ainsi, h pourra être assimilée à une fonction identité ($X = Y$). La fonction d'observation aurait une importance si nous considérions la supervision complète qui inclurait des actions de reconfiguration. En tenant compte de ceci, l'automate S est défini par : $S = \{U, X, X, f, id\}$, où id es la fonction *identité*. Nous pouvons, alors, ne considérer que :

$$S = \{U, X, f\} \tag{32}$$

Où $X = \{X_1, ..., X_i, ...X_K\}$ est l'ensemble d'états fonctionnels et l'ensemble U est celui des événements qui peuvent provoquer des transitions. Dans le contexte de la *surveillance,* ces événements ne se caractérisent souvent que par la constatation de la transition elle-même. Nous appellerons u_{ij} l'événement qui a causé la transition de X_i à X_j. Lorsque la transition est impossible $u_{ij} = 0$; lorsqu'elle correspond à une transition possible dans le modèle de référence, nous posons $u_{ij} = 1$.

Les modèles à états discrets peuvent être considérés selon deux points de vue temporels. Le modèle **asynchrone** consiste à ne considérer le système que par rapport à des événements dont les instants d'occurrence ne sont pas soumis à régularité. Le modèle **synchrone**, que nous considérons ici, suppose une base de temps correspondant au temps d'acquisition des mesures, qui consiste en une séquence infinie d'instants, $\{t_1, t_2, ..., t_i, .t_{i+1}, ...\}$ telle que $t_{i+1} > t_i$. En général cette base de temps sera régulière, c'est-à-dire que :

$$t_{i+1} - t_i = \Delta = constante \tag{33}$$

Nous verrons dans nos applications (§ chapitre 5), à partir de la représentation à base d'épisodes, que cette condition de régularité n'est pas nécessaire.

Remarquons qu'avec un modèle synchrone, la possibilité de permanence dans un même état, entre instants d'échantillonnage, est certaine et donc $u_{ii} = 1 \quad \forall i$.

4.2.1 Machines à états finis (automates) pour la supervision

L'*état* d'un système dynamique est l'information suffisante pour déterminer la trajectoire future du système, pour une fonction d'entrée définie, ou une séquence d'actions. La notion d'***état fonctionnel*** essaie de généraliser le concept d'état à un niveau supérieur. Il s'agit de caractériser la situation dans laquelle se trouve un processus en fonction de son passé, des fonctions qu'il doit exécuter et des possibles transitions vers de nouvelles situations. A cause de cette différence conceptuelle entre l'état fonctionnel et l'état proprement dit d'un système dynamique, il est nécessaire de s'assurer de la sémantique propre à chaque état fonctionnel, pour cela, il convient d'établir un dialogue avec l'expert du processus pour qu'il puisse associer ses connaissances à la définition des états fonctionnels.

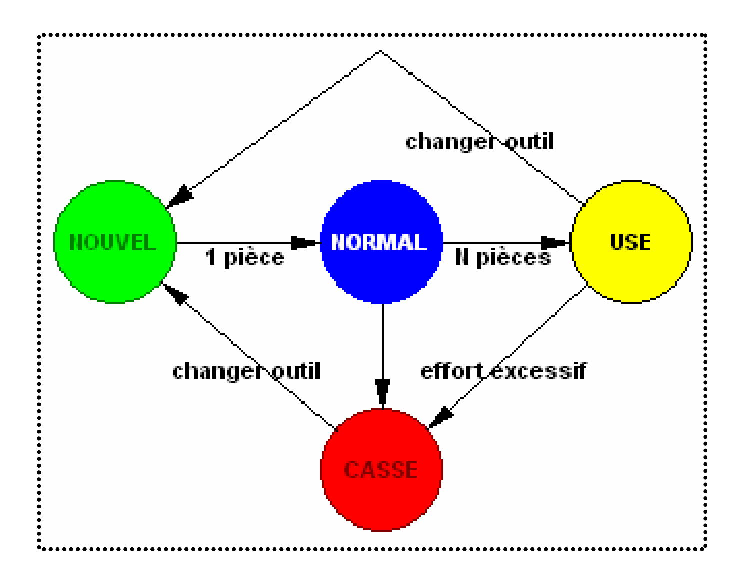


Figure 4.1. Exemple de l'automate pour une machine outil

Prenons l'exemple d'une machine outil. L'état strict a comme composantes la vitesse instantanée de la pièce et les efforts exercés par l'outil dans les trois dimensions spatiales. Cependant, l'état fonctionnel exprimera l'usure de l'outil, la qualité de lubrification, la matière usinée et le type de tâche effectuée. Si l'état de la machine est considéré dans quatre situations ou phases (<nouvel>, <normal>, <usé> et <cassé>), le schéma de la Figure 4.1 représente les transitions possibles entre états fonctionnels et les actions associées.

Nous représentons, de cette façon, le fait que l'outil ne peut pas se casser sans avoir travaillé au moins sur une pièce et, qu'après avoir travaillé sur *N* pièces, il est nécessaire de changer l'outil. Ce schéma représente un automate dont les états fonctionnels correspondent à des situations connues par l'expert. Un système qui est capable de détecter automatiquement dans quel état fonctionnel se trouve le processus et de déterminer quelles sont les transitions favorables à son fonctionnement, réalise une tâche de *surveillance.*

Un système de *détection* pourra vérifier que les transitions qui ont eu lieu sont licites, c'est-à-dire qu'une connexion entre états existe dans le graphe de la Figure 4.1, et ne mène pas à des états inacceptables, comme par exemple l'état d'outil <cassé>.

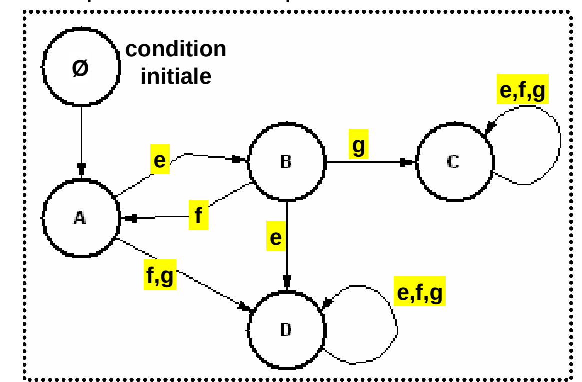
Nous distinguons différents types de transition :

- *les transitions contrôlables :* dans l'exemple, sont contrôlables le changement de l'outil et les pièces qui sont introduites dans la machine,
- *les transitions non-contrôlables* : dans l'exemple, l'effort excessif qui cause la rupture de l'outil est non-controlable.

Le superviseur, en plus de la surveillance, devra proposer des actions qui empêchent d'entrer en conflit, transiter par des états non souhaités et qui, en plus, essaient d'optimiser les critères de qualité de fonctionnement. Le modèle dynamique de l'automate à états finis est une structure appropriée pour la supervision des processus évolutifs.

4.2.2 Matrices de transition

Nous adopterons la représentation sous forme de matrice de la fonction de transition d'état associée au graphe d'états. La Figure 4.2 et le Tableau 4.1 donnent un exemple de cette représentation.



Ensemble d'états :
$X = \{A, B, C, D\}$

Ensemble d'événements :
$U = \{e, f, g\}$

Fonction de transition d'état :
$\{A, B, C, D\} \times \{e, f, g\} \rightarrow f(.,.) \rightarrow \{A, B, C, D\}$

Figure 4.2. Exemple du graphe d'états et transitions

Tableau 4.1. Matrice de fonction de transition d'état

	Evénements de transition		
Etats	e	f	g
A	B	D	D
B	D	A	C
C	C	C	C
D	D	D	D

L'automate de la Figure 4.2 peut générer, par exemple, les séquences suivantes : [ABABABC], [ABABD], [ABABABAD], [ABAD], [ABABABABAB…].Dans le graphe, nous avons ajouté un état fictif Ø pour montrer que A est considéré comme

condition initiale. Les états C et D son absorbants car une action quelconque laisse l'état invariant. En général, l'arc de maintien n'est pas représenté.

Par la suite, nous travaillons avec une représentation dérivée de celle que nous venons de présenter. Cette représentation indique le nombre d'occurrences d'une transition entre un sommet (*état présent*) et les autres états (*états suivants*) possibles. Cette matrice sera notée $\boldsymbol{TB} = \lfloor \boldsymbol{TB_{ij}} \rfloor$ *matrice Brute de Transition*. La matrice du Tableau 4.2 est représentée, pour la séquence [ABABABAD], de la façon suivante :

Tableau 4.2. Exemple de représentation de la Matrice Brute de Transition $\boldsymbol{TB}$

	Etats			
Etats	A	B	C	D
A	--	3	0	1
B	3	--	0	0
C	0	0	--	0
D	0	0	0	--

4.3 Identification des états de l'automate

Dans le chapitre 3, nous avons décrit les moyens d'obtenir les différents états ou modes de fonctionnement du procédé pour l'élaboration d'un *modèle de référence*. Les états significatifs peuvent être identifiés et caractérisés par un apprentissage dirigé ou par un auto-apprentissage selon des connaissances *a priori* des caractéristiques de l'ensemble d'apprentissage. Une autre façon d'identifier les états fonctionnels est à partir de la détection des déviations en ligne, au moyen d'un apprentissage supervisé actif.

Dans le premier cas, nous obtenons les états à partir de l'analyse d'ensembles de données historiques comprenant des enregistrements faits dans des situations connues. Selon les connaissances *a priori* sur les caractéristiques de l'ensemble de données, une partition de l'espace de description sera imposée par l'expert selon l'étiquetage donné à chaque élément. Au contraire, si un auto-apprentissage est effectué, l'algorithme de classification réalise une partition automatique d'après les abstractions faites sur l'ensemble d'apprentissage. Les classes résultantes doivent être interprétées et validées par l'expert jusqu'à obtenir une représentation convenable. Dans le troisième cas, appelé *apprentissage supervisé actif,* il est possible d'identifier des états fonctionnels non présents dans la phase d'apprentissage dirigé. Ceci est fait lors de la détection d'une déviation. Lorsque plusieurs éléments ne sont acceptés dans aucune classe connue (éléments rejetés et assignés à la NIC), ils sont utilisés pour la création de nouvelles classes au moyen d'un apprentissage supervisé actif. De la même façon que pour l'auto-apprentissage, l'expert doit ensuite interpréter et valider ces nouvelles classes comme représentatives d'un ou plusieurs modes de fonctionnement. Une fois la validation faite, les nouveaux états doivent être ajoutés au modèle de référence.

Dans tous les cas, quelque soit le type d'apprentissage utilisé, l'association des classes aux états significatifs est faite directement par l'expert. Il peut associer une classe à un état ou associer plusieurs classes à un même état.

La Figure 4.3 illustre un exemple d'identification des états en opération normale au moyen, d'un auto-apprentissage. Cinq classes on été créées. Le profil des classes résultantes est montré dans la Figure 4.3c). Ces classes ont été interprétées par l'expert de la façon suivante :

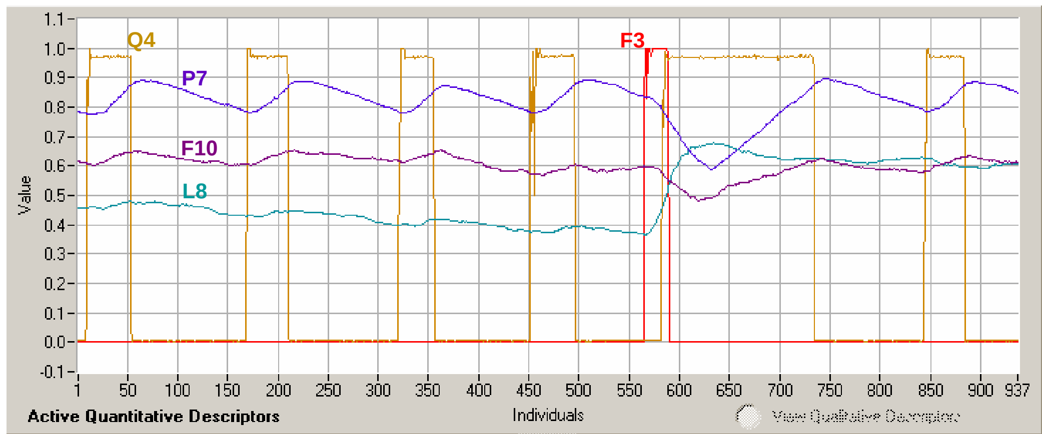
Classe 1 : Le procédé fonctionne normalement, aucune régulation n'est nécessaire

Classe 2 : Régulation de pression – résistance chauffante en fonctionnement

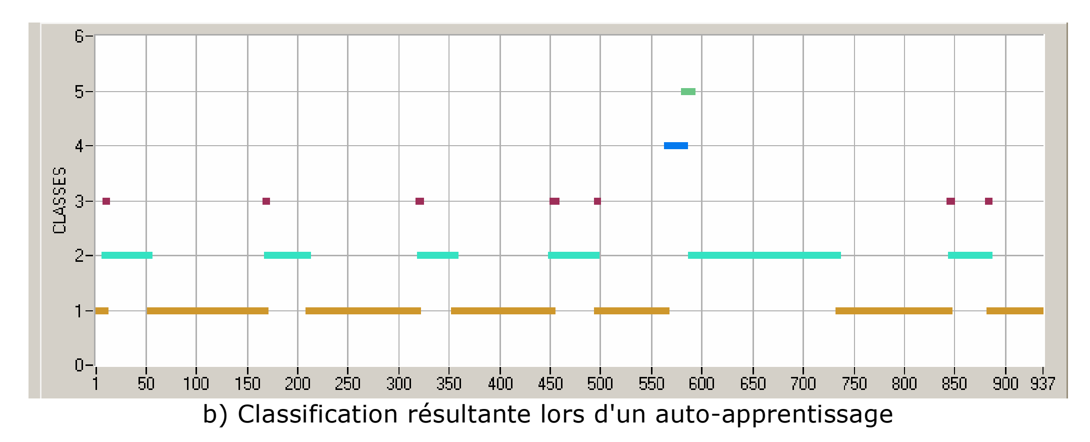
Classe 3 : Transitoire : procédé sans régulation vers régulation de pression

Classe 4 : Régulation du niveau – allumage pompe d'eau

Classe 5 : Régulation du niveau et de pression – allumage pompe d'eau et résistance chauffante en fonctionnement



a) Espace de description avec 5 descripteurs



b) Classification résultante lors d'un auto-apprentissage

Figure 4.3. Exemple de classification pour l'identification des états

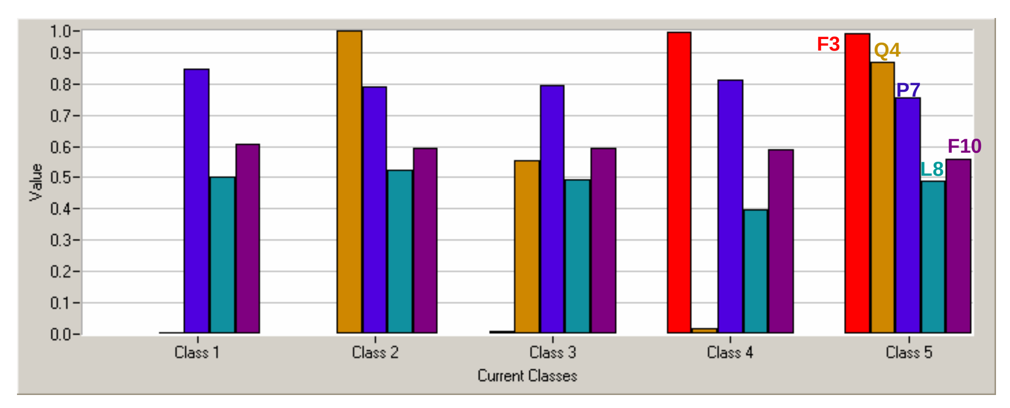


Figure 4.4. Profil des classes

Après l'interprétation des classes générées automatiquement, l'expert a associé les classes à des états représentatifs. Toujours dans le cadre de l'exemple de la Figure 4.3, les états associés sont :

Aucune régulation	Classe 1
Régulation de pression	Classe 2 & 3
Régulation de niveau	Classe 4
Régulation de pression et de niveau	Classe 5

4.4 Identification des transitions à partir de l'apprentissage séquentiel

La construction de l'automate, ou machine à états discrets, qui correspond au modèle de référence évoqué dans les chapitres précédents, comporte l'identification des états et la détermination des transitions entre états.

L'identification des états, nous venons de le voir, s'appuie sur des méthodes de classification appliquées aux données historiques.

En suivant le principe d'identification à partir de données, mais en coordination avec les connaissances des experts, nous proposons ici une méthodologie pour l'identification des transitions.

Dans une première étape, les transitions sont estimées par rapport à leur occurrence. La matrice brute de transitions $\boldsymbol{TB} = \lfloor \boldsymbol{TB}_{ij} \rfloor$ d'états est construite, où $\boldsymbol{TB}_{ij}$ est le nombre de transitions observées de l'état (classe) *i* vers l'état *j*. Ensuite, un outil de dialogue avec l'expert permettra de valider ces transitions et, éventuellement, de les compléter. Cette validation va s'appuyer sur l'interprétation de la matrice brute de transitions qui implique plusieurs constats :

- Tout d'abord, la présence des cases non nulles $\boldsymbol{TB}_{ij} \neq 0$ montre la possibilité de la transition basée sur l'observation.
- Dans le cas fréquent de mesures perturbées par du bruit ou de l'incertitude, nous pouvons considérer qu'une transition est d'autant plus certaine qu'elle a été mesurée souvent; cependant, ceci doit être modulé par le nombre

d'éléments de l'ensemble de départ (TB_{ii}). Il est alors possible de construire la matrice des transitions relatives :

$$TR = [TR_{ij}] = \left[\frac{TB_{ij}}{TB_{ii}}\right] \tag{34}$$

avec $0 \le TR_{ij} \le 1$ et $TR_{ii} = 1$.

Nous pouvons, alors, introduire un seuil de certitude des transitions en dessous duquel une transition sera considérée absente et construire, ainsi, la matrice de transition effective $TE = [TE_{ij}]$

$$\begin{cases} TE_{ij} = 1 & si\ TR_{ij} \ge seuil \\ TE_{ij} = 0 & si\ TR_{ij} < seuil \\ TE_{ii} = 0 \end{cases} \tag{35}$$

La construction de la matrice TE peut :

1. mettre en évidence des oscillations entre deux classes qui, dans certains cas, peuvent traduire une "homogénéité" entre ces classes. Ceci peut, par la suite amener l'expert à regrouper ces deux classes dans un seul état fonctionnel;

2. permettre d'éliminer les transitions non significatives introduites par les erreurs de mesure, ce qui présente l'inconvénient d'éliminer certaines transitions peu fréquentes mais très significatives. Ces transitions peuvent, en effet, correspondre à des pannes qui, en général, ne se présentent pas souvent. C'est pourquoi une validation des résultats par l'expert sera nécessaire, en prenant en compte les spécificités du système surveillé et du scénario d'apprentissage.

Dans le cas où un état j correspond à une de ces situations peu fréquentes, nous avons TB_{ij} petit. Nous pouvons construire alors la matrice de transitions d'arrivée aux états qui permet de mettre en évidence le nombre de fois que l'état est atteint :

$$TA = [TA_{ji}] = \left[\frac{TB_{ji}}{TB_{jj}}\right] \tag{36}$$

Ce type d'information peut être pertinent dans le cas de défaillances particulières non critiques pour lesquelles le processus peut avoir une certaine tolérance. Le processus peut "produire tout de même". Ceci est le cas, notamment, s'il existe une flexibilité matérielle due, par exemple, à une redondance des capteurs.

Nous allons, maintenant, montrer sur un exemple la mise en place des transitions de l'automate et leur validation.

4.4.1 Identification des Transitions

Dans l'approche que nous proposons, étant donné que les observations de la base d'apprentissage suivent un ordre chronologique, les transitions entre classes (états) sont estimées en fonction de leur occurrence dans le scénario d'apprentissage. Ainsi, après un apprentissage, la matrice des transitions ($\boldsymbol{TB}$) entre classes est construite par comptage en considérant chaque changement de classe. Les changements de classe sont dus aux évolutions du DAG et peuvent être analysés en visualisant leur évolution temporelle.

La matrice de transitions indique, d'une part, le nombre d'éléments (instants de mesure) qui ont été attribués à chaque classe (diagonale) et, d'autre part, le nombre de fois qu'il y a eu un passage de la classe C_k (ligne) vers la classe C_l (colonne).

Pour l'exemple, de la Figure 4.3, la matrice brute des transitions $\boldsymbol{TB}$ résultante est donnée par :

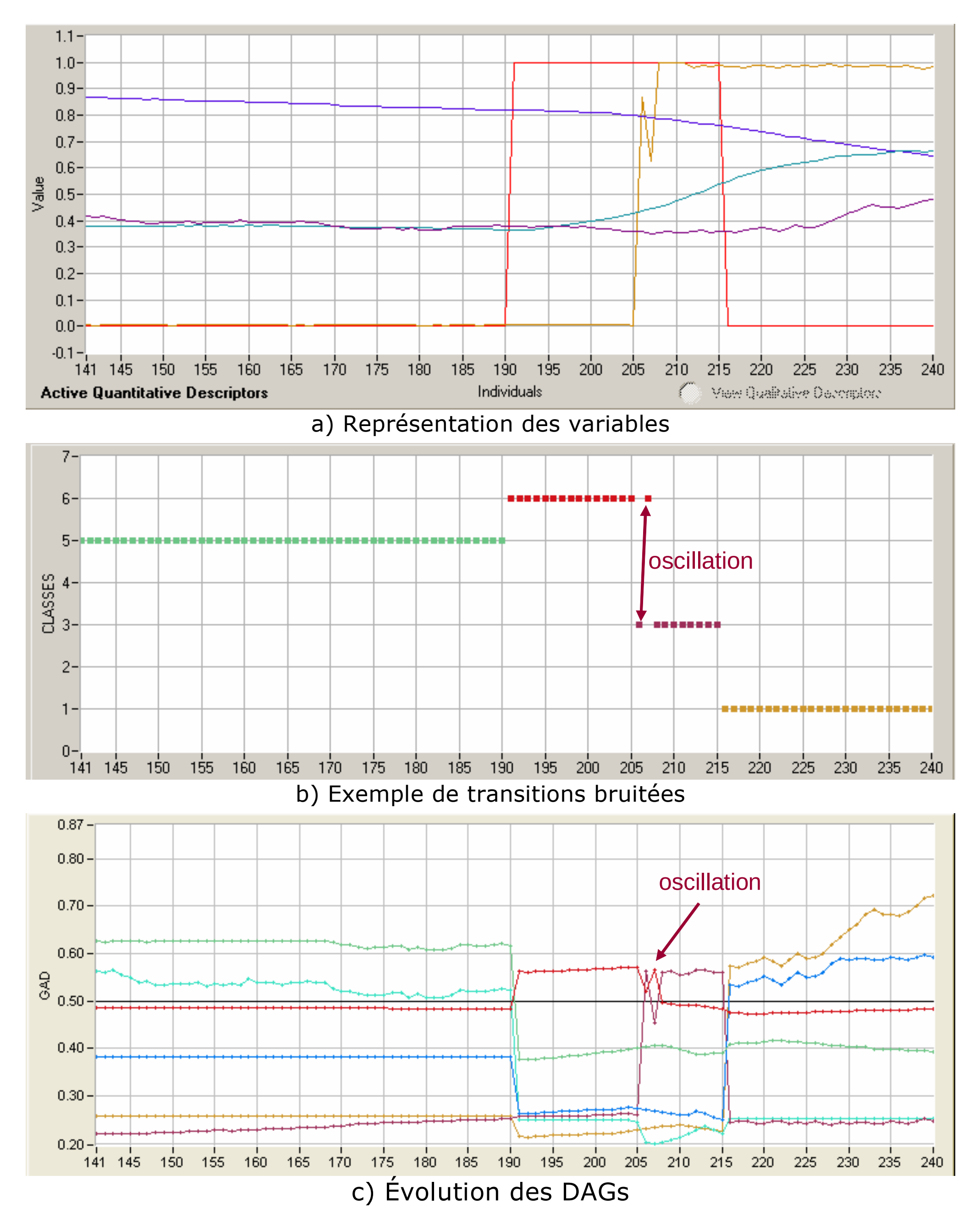
	Class 1	Class 2	Class 3	Class 4	Class 5
Class 1	560	2	3	1	0
Class 2	4	341	5	0	0
Class 3	2	6	11	0	0
Class 4	0	0	0	18	1
Class 5	0	1	0	0	7

Figure 4.5. Matrice de transitions

4.4.2 Validation des Transitions

Une fois que les transitions ont étés obtenues à partir de l'apprentissage, l'opérateur doit les valider et même ajouter des transitions non présentes dans le scénario d'apprentissage mais dont la possibilité de se produire est assurée par la connaissance de l'expert.

Comme le bruit est toujours présent dans l'acquisition des données, il est fréquent que les transitions soient aussi bruitées. C'est-à-dire qu'il y ait une oscillation entre l'appartenance à deux classes. Cette oscillation peut se voir à travers un manque de permanence des éléments dans une classe (voir Figure 4.6). L'opérateur peut décider de ne pas prendre en compte une transition identifiée lors de l'apprentissage s'il s'agit en fait du bruit dans les éléments et que, dans la pratique, ce passage ne doit pas se produire. La validité des transitions peut être faite en s'appuyant sur la matrice de transition effective $\boldsymbol{TE}$. Cependant, l'opérateur ne pourra agir que sur les transitions contrôlables. Sur la Figure 4.6, nous constatons une oscillation réduite à un seul retour de la classe 3 à la classe 6 pour l'individu "206". Nous pouvons nous demander si ce retour est significatif d'une transition réelle de 3 à 6.



a) Représentation des variables

b) Exemple de transitions bruitées

c) Évolution des DAGs

Figure 4.6. Validation des transitions

Pour l'exemple de la Figure 4.5, la matrice ***TR*** peut être une source d'information pour le regroupement de la classe 2 et la classe 3 dans le même état fonctionnel.

L'automate résultant de l'exemple de la Figure 4.3 est illustré dans la Figure 4.7.

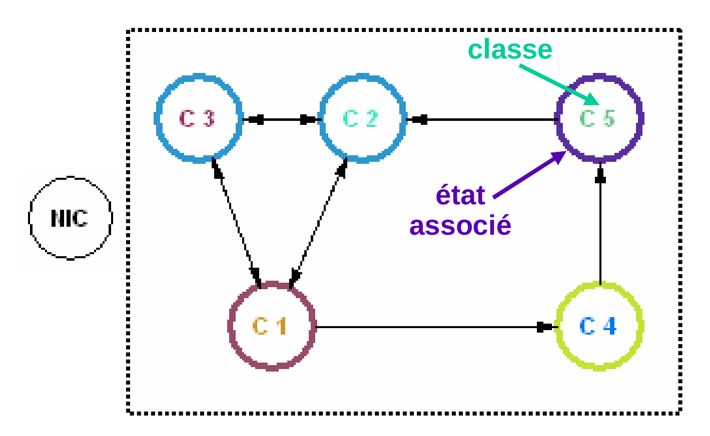


Figure 4.7. Automate résultant de l'apprentissage séquentiel

4.5 Identification des séquences en ligne

En développant cet automate, notre objectif principal était d'aider l'opérateur dans l'identification en ligne des états fonctionnels. Les états étant fixés hors ligne, en ligne, il s'agit plutôt de la reconnaissance des états fonctionnels et de la détection des transitions effectuées par le système ainsi que les événements associés à ces transitions.

4.5.1 Estimation des événements de transition

L'identification de l'automate pour la surveillance, tel qu'il a été défini (§4.2), comporte trois éléments à déterminer :

- les états, obtenus par classification,
- les transitions, déterminées par l'analyse des trajectoires et les connaissances de l'expert,
- les événements qui causent les transitions.

Ces derniers peuvent être des actions externes bien identifiées, comme des actions de contrôle, d'arrêt ou de démarrage de certains sous-systèmes, ou des événements dont l'occurrence est propre au fonctionnement normal ou anormal. Nous traitons ici, indirectement, ces deux cas en ne considérant pas des informations externes aux données, comme par exemple un signal du "conducteur" du procédé.

L'information dont nous disposons dans cette approche est fournie par l'outil SALSA, sous forme de changements détectables dans les valeurs quantitatives ou qualitatives des descripteurs, lorsqu'il y a un changement de classe. Nous cherchons donc à déceler des différences significatives entre les descripteurs de deux classes concernées par une transition. Pour cela, il est nécessaire d'établir un dialogue avec l'expert sur les bases des profils des classes concernées.

Dans le cas de descripteurs quantitatifs, nous avons établi une sémantique simple caractérisant les variations détectées entre les descripteurs des classes, selon

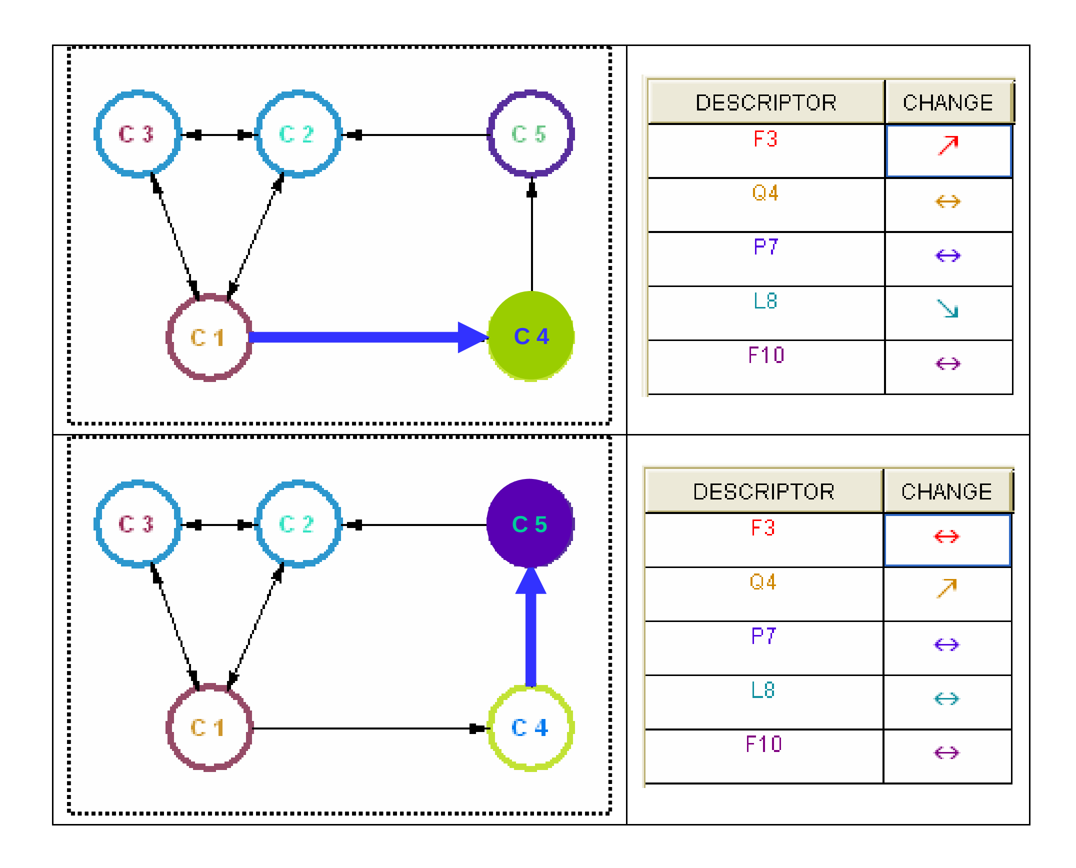
que ceux-ci ont augmenté (↑), diminué (↓) ou sont restés pratiquement inchangés (↔). Dans le cas de descripteurs qualitatifs, l'événement marginal détectable est un changement de modalité. Dans la visualisation de l'automate, nous avons introduit la possibilité de sélectionner graphiquement une transition et dans un tableau associé apparaissent les événements concernant chaque descripteur. Un exemple de cet outil de dialogue est fourni par la Figure 4.8. L'expert peut alors interpréter et valider l'information composée par l'ensemble des événements détectés. Cette information pourra, par la suite, aider l'expert dans une phase de diagnostic en orientant dans un premier temps son analyse vers le ou les descripteurs dont les variations caractérisent le mieux le changement d'état observé.

4.5.2 Présentation à l'utilisateur de l'automate en ligne

Pendant la phase d'exploitation du système, l'opérateur peut visualiser l'état (classe) dans lequel se trouve le processus ainsi que les évolutions des DAGs au fur et à mesure que les observations arrivent. Dès qu'un changement d'état se produit (transition) suite à l'occurrence d'un événement, les attributs qui ont eu une variation importante entre une classe et une autre sont affichés. Cette variation est définie à partir d'un seuil fixe choisi à l'avance. Ainsi, l'opérateur peut suivre l'évolution du procédé selon les changements des variables.

Au niveau de notre outil SALSA, un code couleur a été mis en place pour distinguer les classes des états, ceci correspondant à la fonction ***h*** *d'observation* citée précédemment (§ 4.2).

Dans l'exemple de la Figure 4.8, nous illustrons une séquence d'événements qui produit des changements d'état avec des transitions licites.



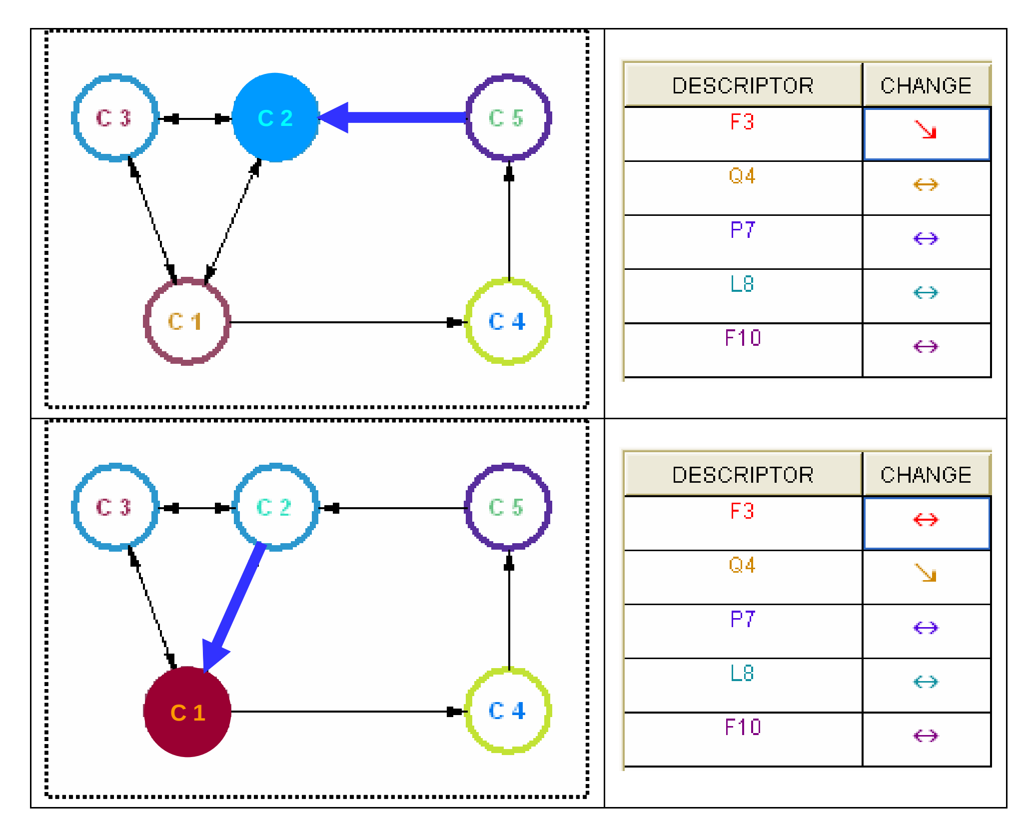


Figure 4.8. Séquence des états connus en ligne

4.5.3 Détection de transitions non valides entre états

Comme nous venons de le voir, les transitions entre états sont obtenues en fonction du scénario qui se déroule mais il est possible que les transitions qui se produisent n'aient pas été rencontrées ou rajoutées par l'expert lors de la phase d'apprentissage (voir Figure 4.9). Ces transitions sont à ce stade dites "non valides". Deux cas peuvent se produire :

- il peut s'agir d'une déviation du comportement qui doit être considérée comme une défaillance dans le processus, ou
- il est possible que les états intermédiaires allant de l'état précédent vers l'actuel n'aient pas été reconnus à cause du bruit ou de la période d'échantillonnage.

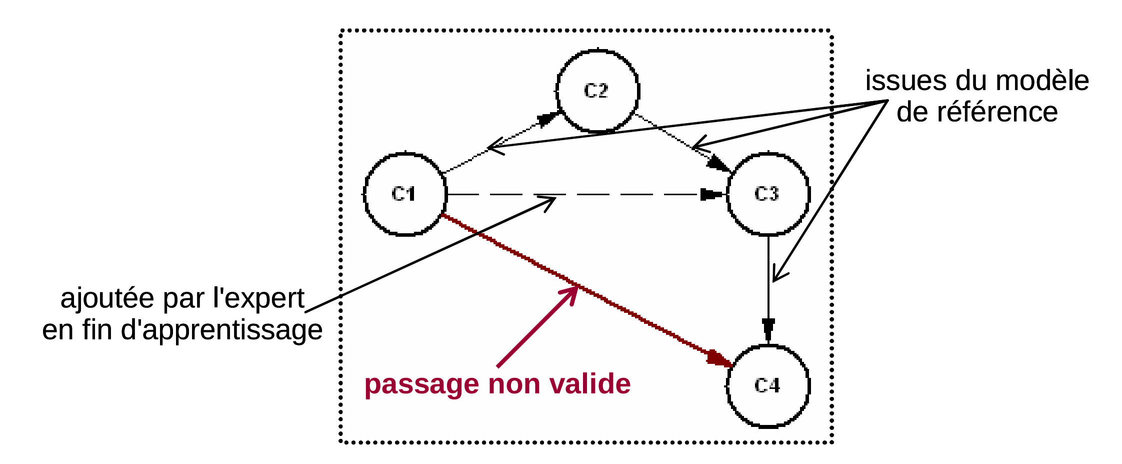


Figure 4.9. Exemple d'une transition non prévue

Quand la transition qui vient de se produire amène vers la classe NIC, c'est-à-dire qu'il y a eu des observations dont les caractéristiques ne conviennent à aucune des classes connues, l'opérateur est alerté de l'événement par une flèche pointillée (voir Figure 4.10). Ceci correspond à la détection d'une déviation.

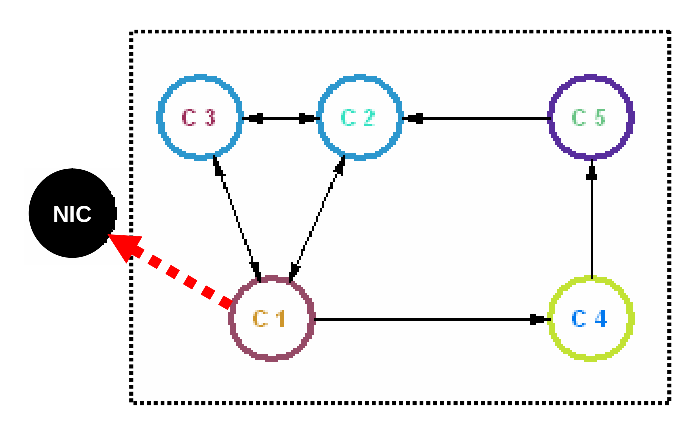


Figure 4.10. Détection d'une déviation

Nous avons montré (cf § 3.4.3) comment cette éventualité peut être exploitée pour compléter le modèle de référence [KEM04].

4.6 Conclusion

Dans ce chapitre, nous avons décrit la façon d'élaborer un modèle à événements discrets (automate) des états fonctionnels du système surveillé, à partir de l'apprentissage séquentiel. Nous avons identifié les moyens d'obtenir les différents états fonctionnels ainsi que les transitions entre ces états.

Cet automate présenté à l'opérateur peut ensuite servir de base pour la surveillance, notamment pour le suivi et la détection des défaillances. Deux types de défaillances peuvent être traités : des défaillances au niveau d'un état ou des défaillances au niveau d'une transition. La présentation à l'expert des descripteurs les

"plus impliqués" dans les transitions observées est une source d'information potentielle pour une phase de diagnostic.

De plus, une utilisation de cet automate à des fins de *pronostic* peut également être envisagée puisqu'en ligne il est possible qu'une évolution vers une situation de défaillance puise se laisser entrevoir au niveau de l'automate. Cette tendance d'évolution peut également être visualisée au travers de l'évolution des DAG (Degré d'Adéquation Global).

Enfin, cet automate peut être une source d'information pour des actions de *reconfiguration,* dans un cadre de supervision.

Chapitre 5

5. SURVEILLANCE D'UNITES PILOTES A L'AIDE DE SALSA

5.1 Introduction

Suite à ce que nous avons exposé, dans les chapitres précédents, sur l'identification des états fonctionnels et l'élaboration d'un modèle de référence pour la surveillance de processus, nous présentons ici les résultats des tests de notre approche à niveau industriel, notamment dans le cadre du projet européen CHEM (§ Annexe A).

La validation des performances de l'outil SALSA a été réalisée sur trois unités industrielles (partenaires du projet CHEM) de nature très différente. Deux de ces unités sont des unités pilotes montées à l'échelle réduite dans un cadre universitaire; il s'agit d'un *gazéifieur de biomasse en lit fluidisé* (Laboratoire de génie chimique à l'UPC de Barcelone) et d'un *générateur de vapeur* pour centrale nucléaire (Laboratoire d'Automatique, Génie Informatique et Signal (LAGIS) de Lille). Le troisième procédé sur lequel les performances de l'outil SALSA ont été testées est un *haut fourneau* (Corus Redcar Blast Furnace au Royaume Uni*)* d'une sidérurgie. Les tests on été réalisés en utilisant des données brutes issues directement des capteurs du procédé ainsi qu'en combinaison avec d'autres outils générant des informations qualitatives et quantitatives.

Pour chaque procédé, les résultats obtenus dans l'étape d'apprentissage hors ligne seront présentés. Dans le cas du générateur de vapeur, nous avons eu la possibilité de valider l'outil SALSA dans l'étape de reconnaissance en ligne. Nous montrerons, dans ce cas, comment une déviation est détectée et caractérisée pour permettre la mise à jour du modèle de référence. Pour le cas du haut fourneau, nous montrerons les résultats obtenus à partir des données traitées par l'outil QUALTRAS (réalisé par le groupe eXiT de l'Université de Girona, Espagne) qui fourni des épisodes qualitatifs et des informations quantitatives.

5.2 Gazéifieur de biomasse en lit fluidisé

Des données de fonctionnement de l'unité pilote pour la gazéification de biomasse en lit fluidisé, de l'Université Polytechnique de Catalogne (UPC) à Barcelone, ont été utilisées pour l'identification des états fonctionnels au travers de l'outil SALSA dans le cadre du projet CHEM.

La **gazéification** est une transformation thermochimique d'un solide combustible (charbon, bois, paille, papier, …) en présence d'un composé gazeux (O_2, air, CO_2, vapeur d'eau…). Le but de cette transformation est, généralement, de convertir le solide en un mélange gazeux. La gazéification se décrit, du point de vue chimique, comme une réaction endothermique, hétérogène, entre le carbone (C) contenu dans le solide et un gaz "réactant" qui peut être la vapeur d'eau (H_2O) ou du dioxyde de carbone (CO_2).

$$C + H_2O \rightarrow CO + H_2 \quad (37)$$

$$C + CO_2 \rightarrow 2CO \quad (38)$$

Sur le plan industriel, l'objectif est de favoriser les réactions (37) et (38), qui vont produire le gaz combustible. Pour ce faire, il faut préalablement apporter les éléments nécessaires à ces deux réactions : le charbon, très concentré en carbone, les réactants CO_2 et H_2O ainsi qu'une quantité importante d'énergie [VAN03].

La gazéification permet d'accéder à des solutions techniques de production d'électricité à fort rendement, à la fois pour les petites et moyennes puissances au moyen de la combustion des gaz dans des moteurs thermiques, ou les grandes puissances, avec l'utilisation des turbines à gaz. De plus, lorsque la biomasse est concernée, à l'enjeu énergétique s'ajoute un enjeu environnemental : il s'agit de la substitution des énergies fossiles et de la réduction des émissions de gaz à effet de serre.

Le gazéifieur, qui se trouve à l'UPC, est un réacteur chimique qui produit de l'hydrogène à partir de charbon ou de déchets de bois (biomasse) et de vapeur d'eau (gazogène). Un schéma du gazéifieur est donné par la Figure 5.1. Le Tableau 5.1 montre les références des capteurs. Le réacteur fonctionne en lit fluidisé à 930ºC et 1 atm. Le charbon (ou la biomasse) est introduit par une vis rotative située au pied du réacteur et commandée électroniquement. L'alimentation en air plus vapeur d'eau (agent de gazéification) est faite par un distributeur en pied. En tête, le produit récupéré est un mélange d'hydrogène (H_2) et d'oxyde de carbone (CO). Sa composition est mesurée indirectement par un contrôle continu en ligne des concentrations de dioxyde de carbone (CO_2) et d'oxygène (O_2).

Le gazéifieur est équipé de deux sondes de pression, de deux débitmètres, de quatre thermocouples et d'un analyseur en ligne. Il a également des déclencheurs pour le système de chauffage, la vis rotative et les pompes. Il opère en mode continu.

La qualité du produit à la sortie (émissions de CO) commande l'opération du gazéifieur. Néanmoins, à cause des dimensions réduites du réacteur, le processus de

gazéification est difficile à opérer et à contrôler et la stabilisation est difficile à atteindre.

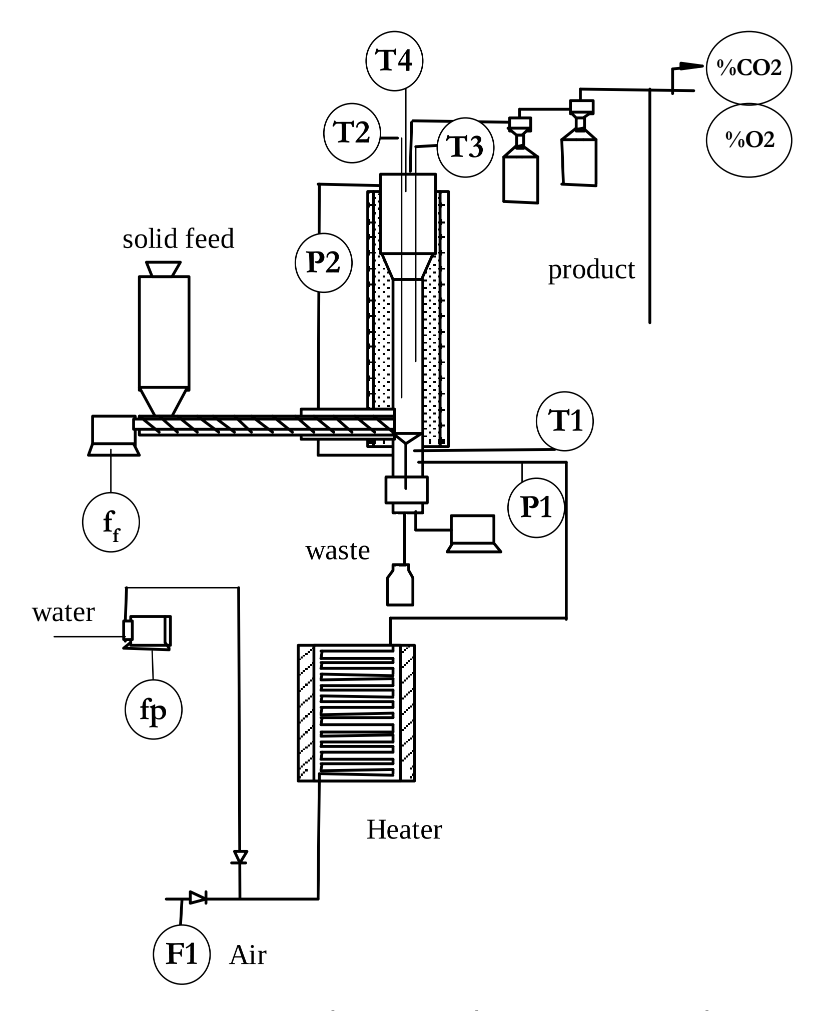


Figure 5.1. Schéma du gazéifieur en lit fluidisé

Tableau 5.1. Liste des capteurs du gazéifieur

Nom capteur	Variable mesurée
F1	Débit d'air
Fp	Fréquence de la pompe à eau
P1	Pression (entrée des gaz)
P2	Pression différentielle
T1	Température à l'entrée du réacteur
T2	Température au fond du réacteur
T3	Température dans le réacteur
T4	Température à l'extérieur du réacteur
ff	Fréquence de la vis rotatoire
$\%CO_2$	Concentration de dioxyde de carbone dans le gaz
$\%O_2$	Concentration d'oxygène dans le gaz

5.2.1 Données brutes du fonctionnement

Nous disposons uniquement des données historiques du réacteur, contenant des défaillances qui ont eu lieu dans le passé. Les mesures de pression de gaz à l'entrée ainsi que la pression différentielle du réacteur n'ont pas été considérées car les lectures des capteurs n'étaient pas fiables. Donc, les variables brutes disponibles et celles que nous avons utilisées pour définir l'espace de description sont les suivantes: la température à l'entrée du réacteur (T1), la température au fond du réacteur (T2), la température dans le réacteur (T3) et les concentrations de dioxyde de carbone (CO_2) et d'oxygène (O_2). L'ensemble de données disponible pour réaliser l'apprentissage est illustré dans Figure 5.2.

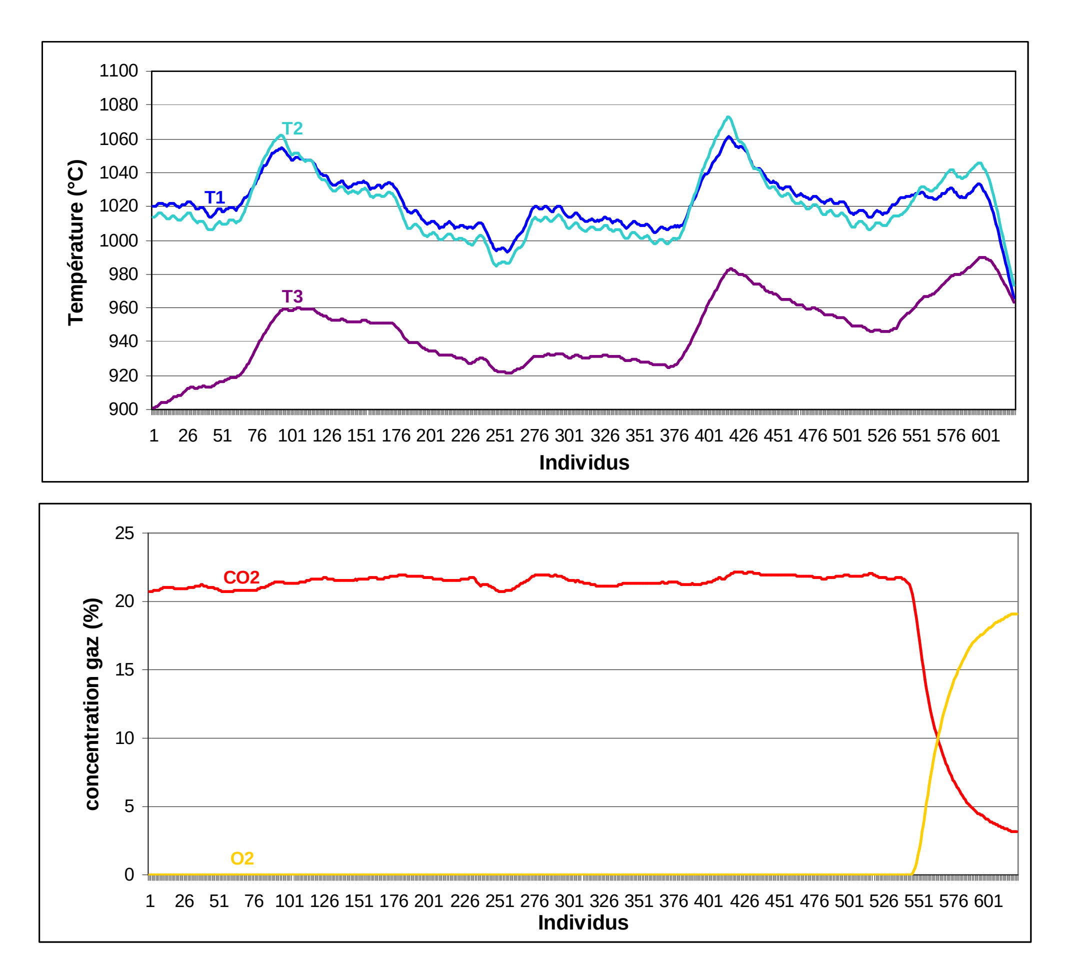


Figure 5.2. Ensemble de données brutes du gazéifieur

5.2.2 Résultats à partir de données brutes

A partir de l'ensemble des données brutes enregistrées lors d'un fonctionnement, nous avons utilisé comme espace de représentation les descripteurs et les valeurs pour la normalisation montrés dans le Tableau 5.2. La Figure 5.3 illustre l'ensemble des données utilisé pour l'identification des états fonctionnels du gazéifieur.

Tableau 5.2. Configuration de l'espace de représentation (données brutes)

Descripteur	Valeurs	
	Maximale	Minimale
$\%CO_2$	22.8	2.8
$\%O_2$	19.5	0
T1 (°C)	1070	950
T2 (°C)	1080	950
T3 (°C)	1000	900

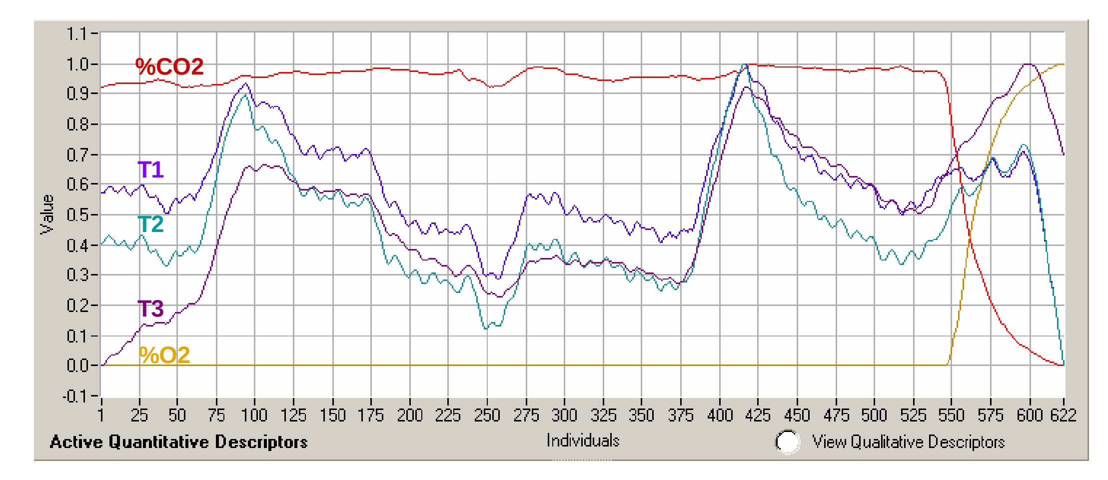
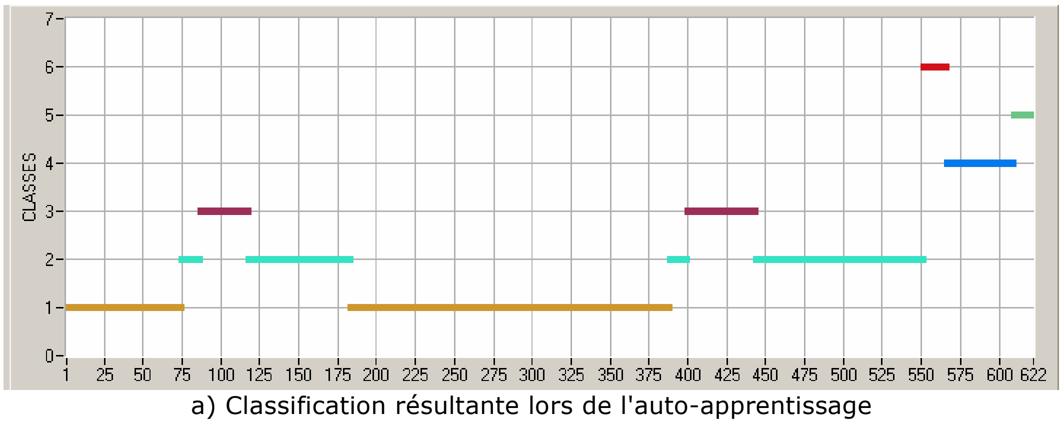
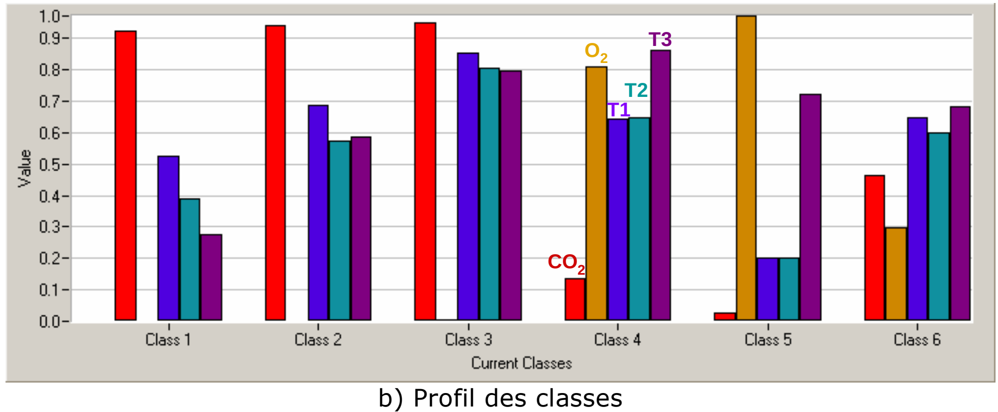


Figure 5.3. Ensemble d'apprentissage (données brutes)



a) Classification résultante lors de l'auto-apprentissage



b) Profil des classes

Figure 5.4. Classification pour l'identification des états du gazéifieur

Pour l'identification des états fonctionnels du gazéifieur présents dans l'ensemble d'apprentissage, nous avons réalisé un auto-apprentissage. Dans [KEM03] les auteurs ont montré plusieurs partitions pour différentes valeurs de l'indice d'exigence ainsi que la partition résultante à partir d'un apprentissage dirigé. La classification présentée dans la Figure 5.4 a été obtenue en utilisant la fonction *binomiale centrée* (équation (14)) pour le calcul des adéquations marginales (DAM). La famille des connectifs choisie pour obtenir l'adéquation globale (DAG) est le *Minimum-Maximum* avec un indice d'exigence $\alpha = 0.9$. A partir de ces paramètres, 6 classes ont été obtenues. Ces classes ont été interprétées par l'expert selon ses connaissances et à l'aide du profil des classes (Figure 5.4b)) ainsi que du HAZOP disponible; elles ont été associées aux cinq situations suivantes :

SITUATION	**Description**	**Classe**
Opération normale	Températures suivent le point d'opération (stabilité) Combustion normale	*C1*
Haut débit d'eau	Augmentation des températures notamment T2 Combustion normale (%CO_2 maximale)	*C3*
Blocage à l'entrée du solide/ obstruction fond du réacteur	Incrément des températures (T2 et T3), combustion arrêtée (%CO_2 bas et incrément %O_2)	*C4*
Arrêt du réacteur	Diminution des températures, combustion arrêtée	*C5*
Transitoire ou alarme	Vers une situation de défaillance ou retour en opération normale. Changement des températures et des compositions du gaz de sortie (%CO_2 & %O_2)	*C2* & *C6*

Selon l'évolution des DAGs de la Figure 5.5 et en considérant la matrice de transition, l'automate de la Figure 5.6 a été construit. Nous constatons qu'après une augmentation du débit d'eau pour l'obtention de la vapeur d'eau, il est possible pour le système de revenir à l'état stable. Par contre, lorsqu'il y a une obstruction au niveau de l'entrée du solide (charbon ou biomasse), le système ne peut pas revenir au fonctionnement normal et doit être arrêté. L'expert a ajouté une transition (transition en pointillé dans la Figure 5.6) dès l'opération normale (*C1*) vers le blocage à l'entrée du solide (*C4*) car même, si cette transition n'est pas présente dans le scénario d'apprentissage, la connaissance du système permet de dire qu'elle est possible

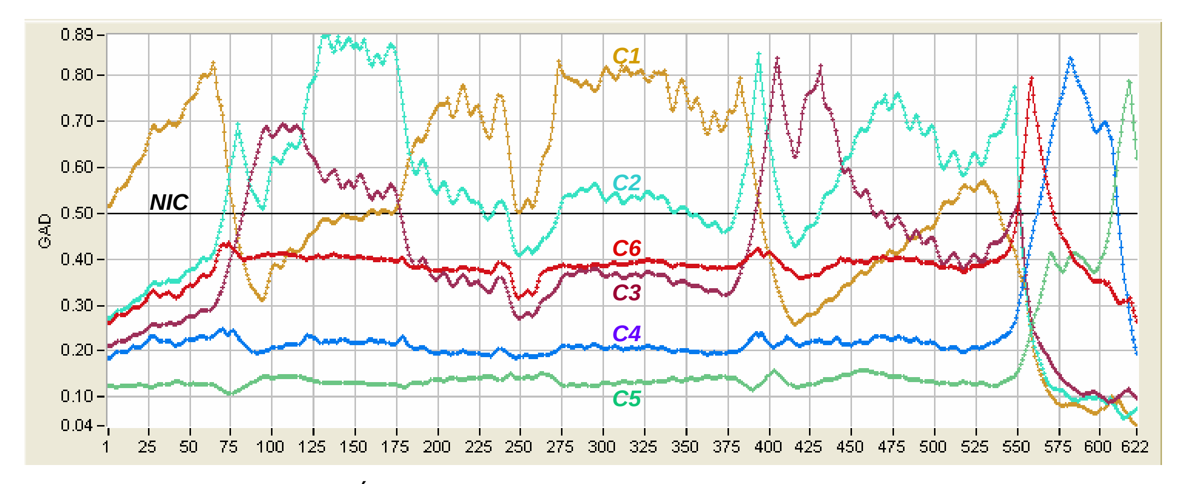


Figure 5.5. Évolution des DAGs pour la classification résultante

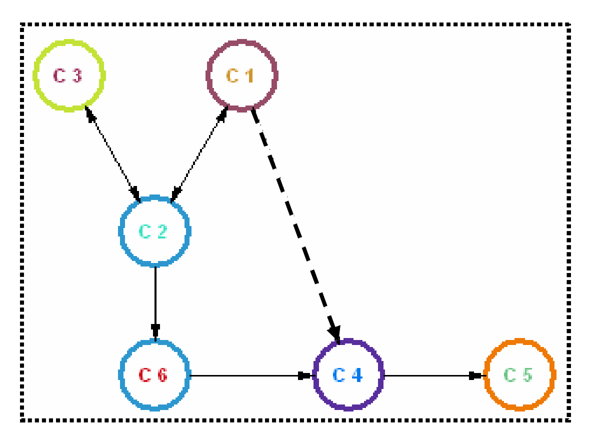


Figure 5.6. Automate pour la gazéification (données brutes)

5.2.3 Données transformées en épisodes

5.2.3.1 Représentation à base d'épisodes

Le mode en opération normale du gazéifieur ne peut pas garantir un fonctionnement autour d'un point d'opération fixe car il existe des variables non contrôlables telles que l'humidité et la température ambiante qui changent les conditions d'opération du réacteur. De plus, la quantité de charbon (ou biomasse) à l'entrée du réacteur n'est pas uniforme et, donc, selon la qualité de la composition du gaz à la sortie, l'opérateur modifie le débit d'eau (Fp) et le débit d'air (F1) pour garantir la qualité du gaz. Étant données ces conditions opératoires, l'identification des états représentatifs du processus à partir des données brutes reste restreinte aux conditions spécifiques du scénario qui se déroule. L'utilisation des tendances qualitatives des variables, au moyen de la représentation en épisodes, peut être une alternative pour résoudre le problème du point d'opération variable. De ce fait, et dans le cadre du projet CHEM, nous avons travaillé en collaboration avec l'Université de Girone (UdG), de manière à extraire des informations qualitatives et quantitatives à partir de la représentation des épisodes des variables du processus. La séquence d'épisodes est générée par l'outil QUALTRAS. Ces informations peuvent, ensuite, être utilisées dans l'identification des situations représentatives par le biais de la classification avec SALSA.

Les épisodes ont été construits uniquement à partir de la première dérivée de chaque signal; 7 types d'épisodes ont été identifiés dans cette approche. Leur représentation est donnée dans la Figure 5.7.

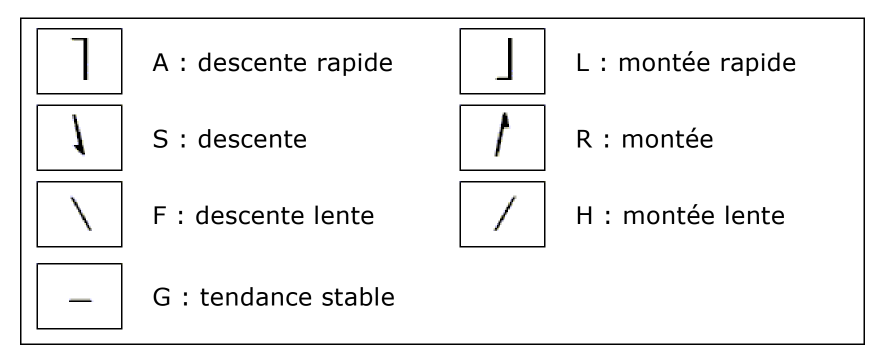


Figure 5.7. Type d'épisodes pour les variables du gazéifieur

Un exemple de représentation à base d'épisodes pour les variables du gazéifieur est montré dans la Figure 5.8, où nous observons les tendances pour les compositions de dioxyde de carbone (CO_2) et d'oxygène (O_2).

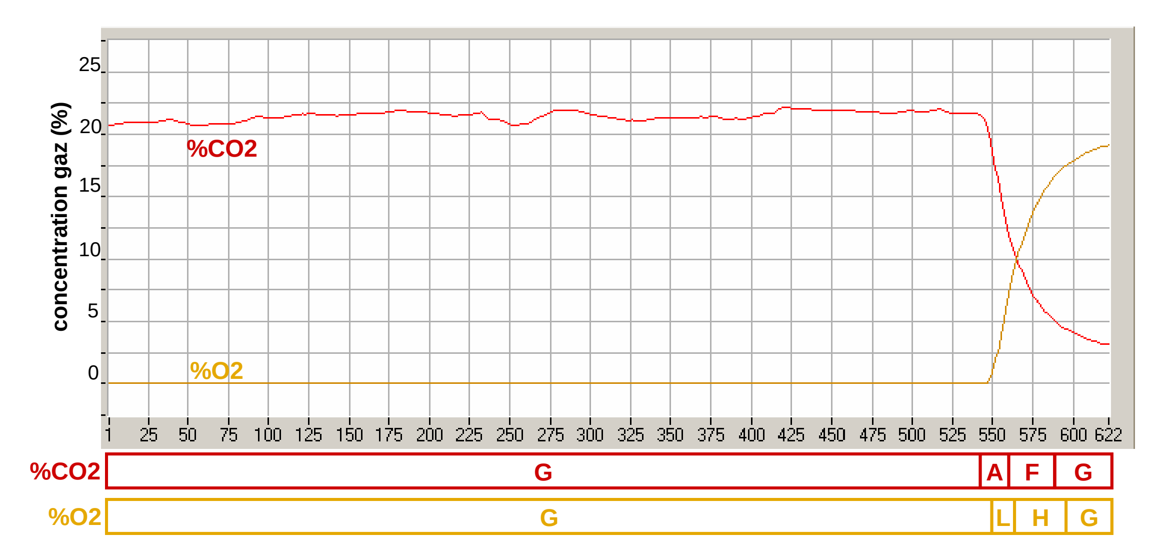


Figure 5.8. Exemple de représentation des variables avec les épisodes

En prenant en compte le fait que les épisodes sont asynchrones par nature et pour permettre la classification en utilisant plusieurs variables en même temps, une procédure de synchronisation a été introduite afin d'adapter les informations fournies par la représentation des épisodes à notre approche. Ainsi, des informations telles que le type d'épisode courant, le type d'épisode précédent, la durée de l'épisode, la valeur initiale de la variable dans chaque épisode, la pente de l'épisode, sont utilisées pour construire un nouvel élément de l'ensemble d'apprentissage à chaque instant qu'un changement d'épisode se produit. Les éléments à classer ou à reconnaître sont les instants où il y a eu un changement d'épisode dans au moins une variable. Pour l'exemple de la Figure 5.8, nous avons obtenu l'ensemble d'éléments suivants :

Tableau 5.3. Ensemble d'éléments issus des épisodes pour les compositions CO_2 et O_2

Instant	**%CO2**			**%O2**		
	Episode courant	Episode précédent	Valeur finale	Episode courant	Episode précédent	Valeur Finale
1	G	G	21	G	G	0.5
547	A	G	10.06	G	G	0.5
550	A	G	10.06	L	G	10.9
565	F	A	5.24	L	G	10.9
568	F	A	5.24	H	L	13.38
585	G	F	3.09	H	L	13.38
592	G	F	3.09	G	H	19.2

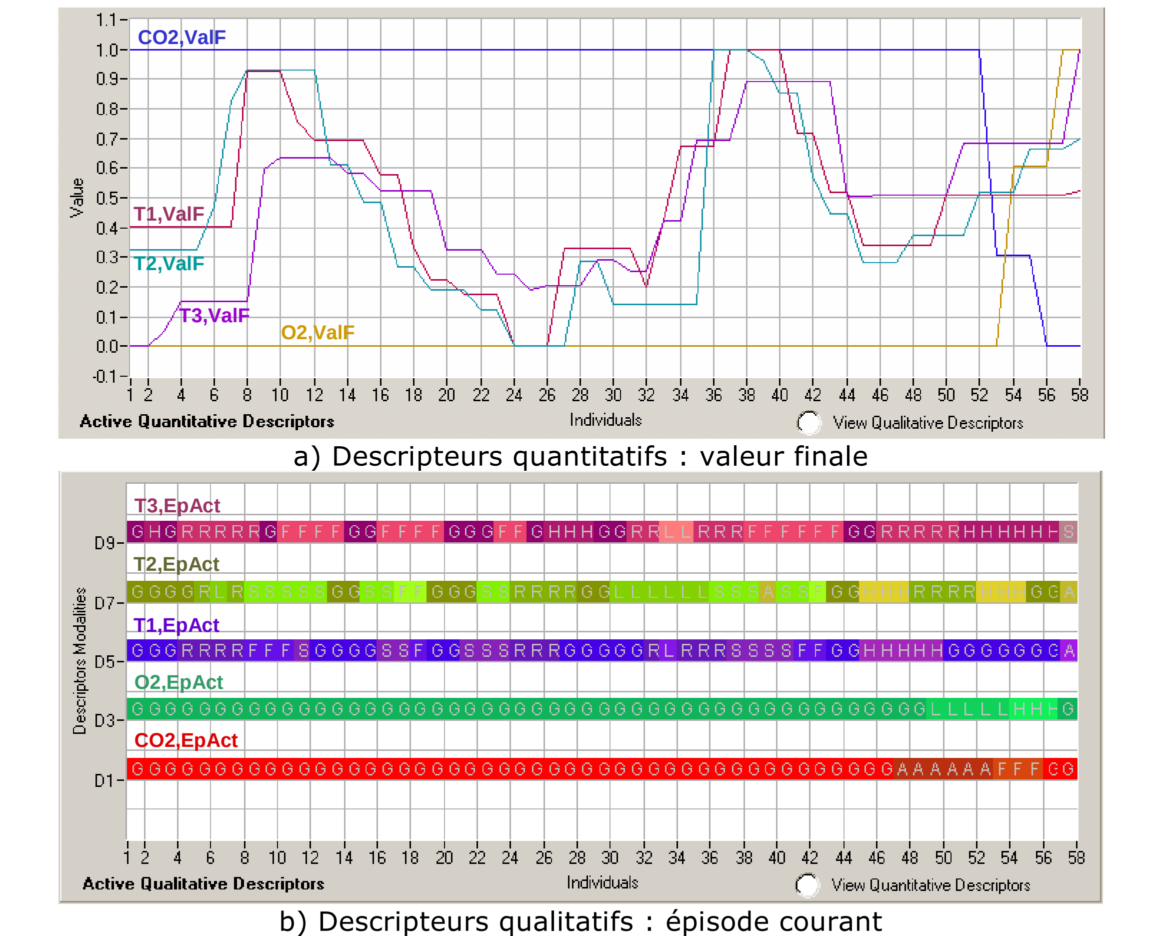
Les descripteurs, comme le montre le Tableau 5.3, sont les épisodes présents, sous forme qualitative, et les valeurs finales des épisodes précédents, sous forme numérique. Ces informations sont répétées lorsqu'il n'y a pas eu de changement dans le descripteur correspondant.

5.2.3.2 Résultats à partir des données mixtes (épisodes)

Les descripteurs utilisés pour l'apprentissage sont des informations qualitatives (type d'épisode) et quantitatives (valeur finale de l'épisode précédent). Ces informations sont issues de la représentation à base d'épisodes des variables du processus. Les variables prises en compte pour l'apprentissage sont : %CO_2, %O_2, T1, T2 et T3. Pour chaque variable, deux descripteurs ont été utilisés : l'épisode courant (EpAct) de type qualitatif et la valeur finale (EpVal) de type numérique. Dans le Tableau 5.4 une description des descripteurs utilisés est donnée. La Figure 5.9 illustre l'ensemble d'apprentissage obtenu à partir de la représentation des épisodes.

Tableau 5.4. Configuration de l'espace de description avec épisodes

DESCRIPTEURS	TYPE	VALEURS	
CO2,EpAct	Qualitatif	Modalités : G, A, F	
CO2,ValF	Quantitatif	Min =5.24	Max =21
O2,EpAct	Qualitatif	Modalités : G, L, H	
O2,ValF	Quantitatif	Min =0	Max =17.66
T1,EpAct	Qualitatif	Modalités : G, R, F, S, L, H, A	
T1,ValF	Quantitatif	Min =997	Max =1055
T2,EpAct	Qualitatif	Modalités : G, R, L, S, F, A	
T2,ValF	Quantitatif	Min =989	Max =1062
T3,EpAct	Qualitatif	Modalités : G, H, R, F, L, S	
T3,ValF	Quantitatif	Min =907	Max =990

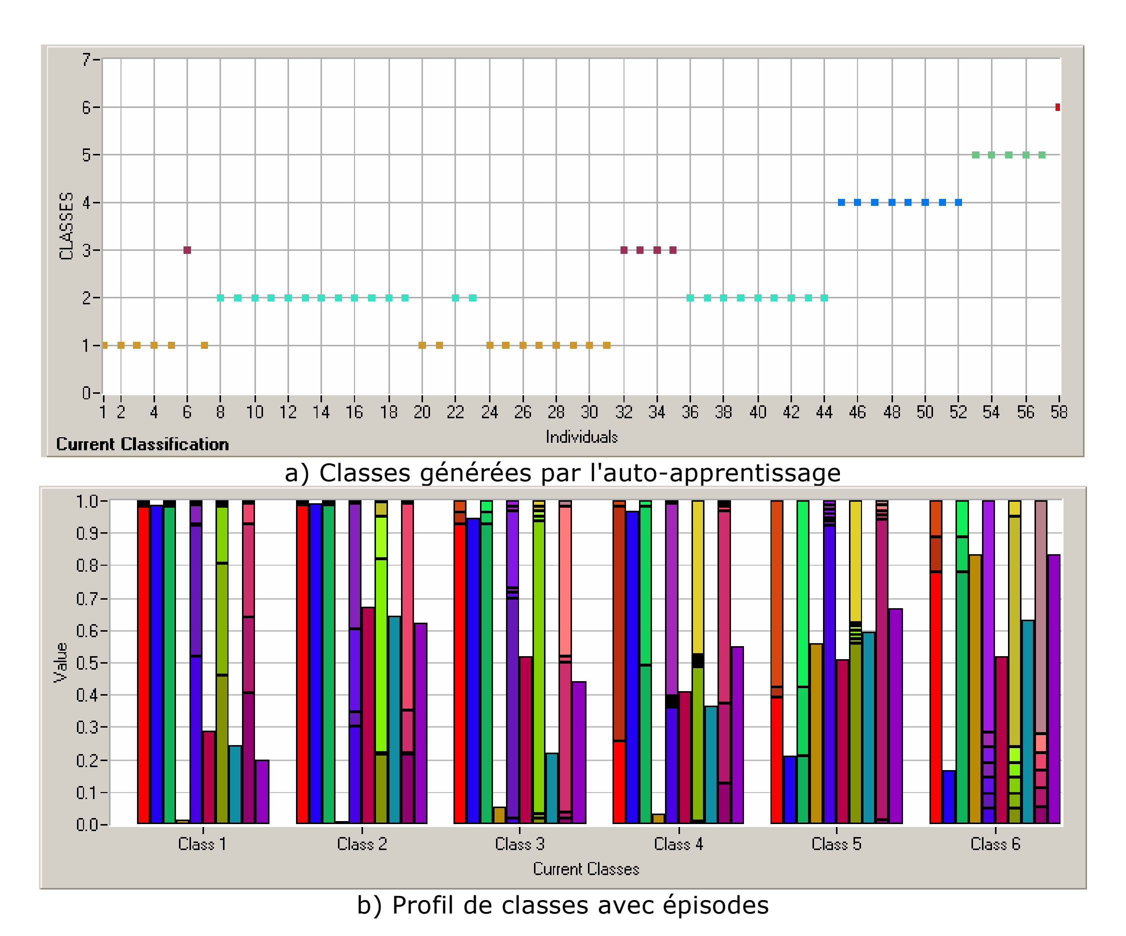


a) Descripteurs quantitatifs : valeur finale

b) Descripteurs qualitatifs : épisode courant

Figure 5.9. Ensemble d'apprentissage (épisodes)

Comme dans le cas des données brutes pour l'élaboration du modèle de référence, un auto-apprentissage a été réalisé. En ce qui concerne le réglage des paramètres de LAMDA, la meilleure partition a été obtenue en choisissant la fonction *gaussienne* (équation (17)) pour le calcul des DAM pour les descripteurs quantitatifs, la famille du *produit*, comme opérateur d'agrégation, pour le DAG avec un indice d'exigence $\alpha = 1$. La Figure 5.10 illustre la classification résultante où 6 classes ont été générées. Les descripteurs apparaissent dans l'ordre suivant : CO2-EpAct, CO2-ValF, O2-EpAct, O2-ValF, T1-EpAct, T1-ValF, T2-EpAct, T2-ValF, T3-EpAct, T3-ValF.



a) Classes générées par l'auto-apprentissage

b) Profil de classes avec épisodes

Figure 5.10. Classification avec épisodes et auto-apprentissage

Les situations qui ont été identifiées d'après la classification et selon les séquences des épisodes pour chaque variable sont présentées dans le Tableau 5.5. L'automate résultant à partir de la classification est illustré dans la Figure 5.11. En comparant cet automate avec celui obtenu avec les données brutes (Figure 5.6), nous remarquons une adéquation parfaite. Comme pour l'automate généré à partir des données brutes, l'opérateur a rajouté une transition de la classe *C1* (opération normale) vers la classe *C5* (blocage à l'entrée du solide). Nous retrouvons, aussi, les mêmes séquences dans l'évolution du comportement du processus. En effet, lorsqu'une obstruction au fond du réacteur (défaillance) est détectée, le procédé doit être arrêté car il s'agit d'une situation critique. Cependant, la classification issue des épisodes a permis d'identifier l'intervention de l'opérateur qui a diminué le débit d'eau, ce qui se traduit par une baisse de température (individus 22 et 23 dans la Figure 5.9).

Nous pouvons remarquer que l'utilisation des épisodes a réduit le nombre d'éléments à classer et que l'écart temporel entre eux n'est pas constant. Ainsi, l'intervention de l'opérateur dont nous venons de parler se situé initialement sur la période *t=230* à *t=275* (Figure 5.3).

Tableau 5.5. Situations identifiées à partir de la classification (épisodes)

SITUATION / CLASSE	Variable	épisodes
Opération Normale *C1*	%CO2	G
	%O2	G
	T1	G
	T2	G, R
	T3	G, H, R
Haut débit d'eau *C3*	%CO2	G
	%O2	G
	T1	R, L
	T2	L
	T3	R, L
Transitoire : retour opération normale *C2*	%CO2	G
	%O2	G
	T1	F, S, G
	T2	S, G, F
	T3	G, F
Alarme vers situation anormale *C4*	%CO2	G, A
	%O2	G, L
	T1	H, G
	T2	G, H, R
	T3	G, R
Obstruction fond du réacteur / Blocage à l'entrée du solide *C5*	%CO2	F, G
	%O2	L, H, G
	T1	G
	T2	H, G
	T3	H
Arrêt du réacteur *C6*	%CO2	G
	%O2	G
	T1	A
	T2	A
	T3	S

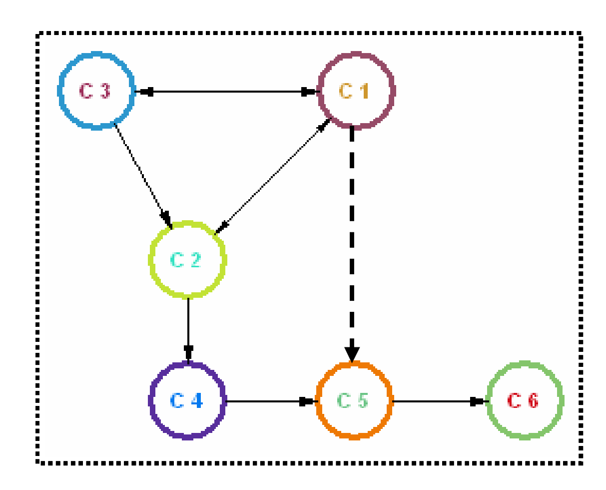


Figure 5.11. Automate généré à partir d'épisodes

5.3 Générateur de vapeur (*LAGIS)*

L'unité de test a été conçue comme une reproduction à échelle pilote d'une unité de production de vapeur pour une centrale nucléaire (voir Figure 5.12). Cette unité se situe à l'Université de Lille, dans le Laboratoire d'Automatique, Génie Informatique & Signal (LAGIS).



Figure 5.12. Aperçu du générateur de vapeur

Le générateur de vapeur est un procédé complexe, non linéaire qui reproduit les mêmes phénomènes thermodynamiques qu'un procédé industriel réel. Le schéma général du générateur est illustré par la Figure 5.13. Les caractéristiques des capteurs et des actionneurs sont présentées dans le Tableau 5.6. L'installation a été construite autour de 4 sous-systèmes :

- un receveur avec système d'alimentation en eau fraîche,
- un bouilleur équipé d'une résistance chauffante de 60kW,
- un système de production de vapeur,
- un condenseur avec échangeur thermique.

Le courant d'eau d'alimentation (*F3*) circule via 2 pompes d'alimentation fonctionnant en parallèle. Chaque pompe est contrôlée par un régulateur de type tout-ou-rien, pour maintenir constant le niveau (*L8*) de l'eau dans le générateur de vapeur (bouilleur). La puissance de chauffage *Q4* est déterminée sur la base de la pression *P7* dans l'accumulateur. La détente de la vapeur générée est réalisée au travers de 3 vannes fonctionnant en parallèle (en l'absence de la turbine). Une vanne (*V12*) de by-pass manuelle permet de simuler le passage direct de la vapeur vers le condenseur. La vanne *V2* est automatiquement contrôlée pour maintenir une valeur adéquate de la

pression *P15* dans le condenseur. Dans une installation industrielle, la vapeur produite alimenterait une turbine pour produire de l'électricité mais, dans l'installation pilote, la vapeur est condensée et renvoyée dans un bac receveur pour être réutilisée, ensuite, dans le générateur de vapeur.

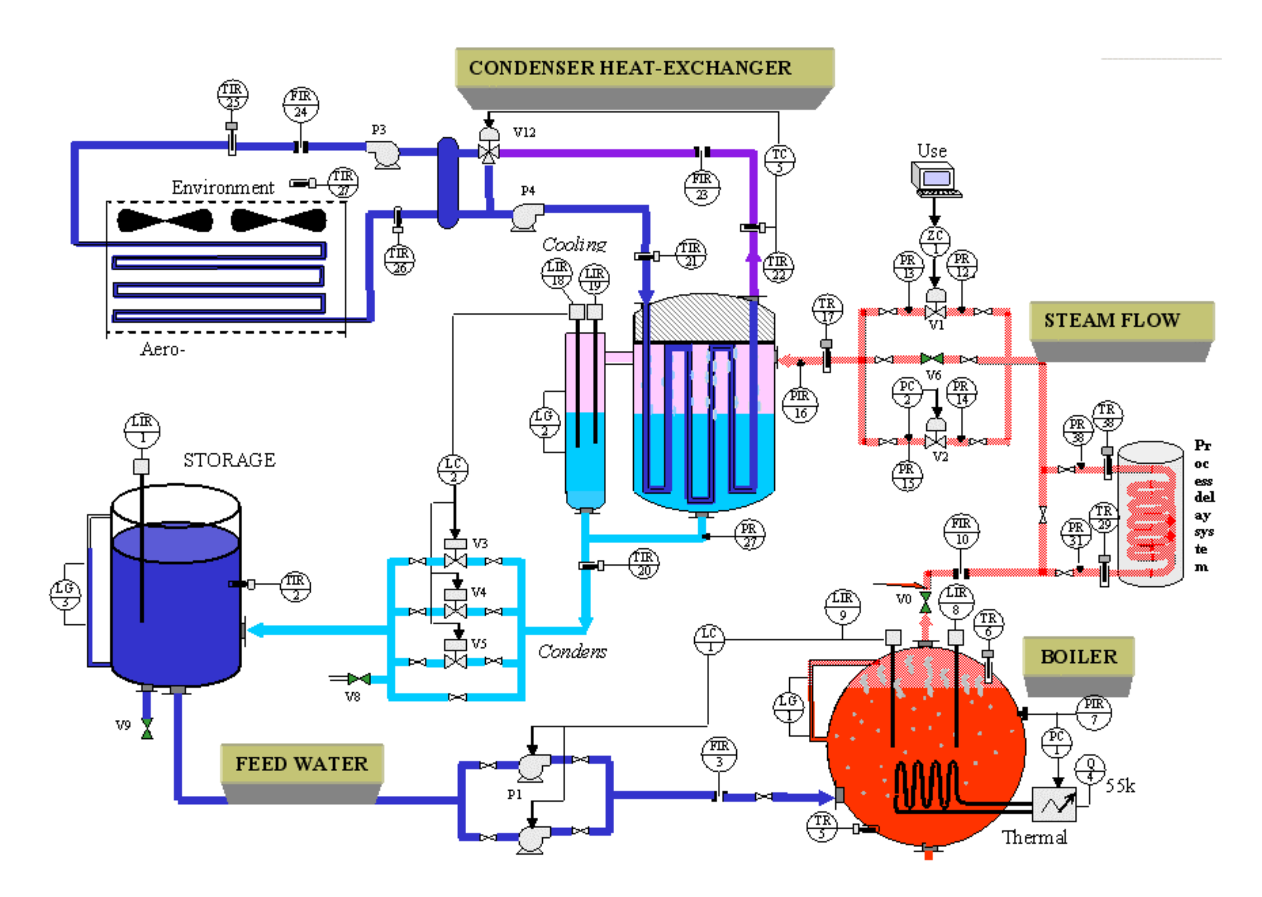


Figure 5.13. Schéma général de l'installation

Tableau 5.6. Caractéristiques des capteurs et actionneurs

Symbole	Descriptif	Unités	Rang	Point d'opération
L1	Niveau du receveur	L	0-420	200
T2	Température d'eau du receveur	°c	0-100	37
F3	Débit d'eau d'alimentation	L/h	0-1600	950
Q4	Puissance de chauffage	Kw	0-60	60
T5	Température de l'eau dans bouilleur	°c	0-200	168
T6	Température de la vapeur dans bouilleur	°c	0-200	170
P7	Pression dans bouilleur	Bar abs	0-16	8
L8	Niveau d'eau dans bouilleur	L	143-156	146
F10	Débit de la vapeur en sortie du bouilleur	Kg/h	0-100	83
V1	Vanne à commande continue	%	0-100	
V2	Vanne à commande continue	%	0-100	
V12	Vanne commandée à trois voies			
L18	Niveau d'eau dans le condenseur	Mm	0-400	300
T27	Température ambiante	0-100	°c	20
V3,V4,V5	Vannes de décharge tout-ou-rien			

En régime normal d'opération, la pression (*P7*) dans le bouilleur est régulée par un contrôleur "tout-ou-rien" qui agit sur la résistance chauffante pour maintenir la

pression entre ±0.2 bars du point d'opération. De même, le niveau (*L8*) d'eau est régulé entre ±3 litres du point d'opération en agissant sur la pompe (*P1*). Le niveau d'eau dans le condenseur est maintenu entre ±0.5 litres du point de consigne à travers des vannes de décharge (V3, V4, V5).

Étant donné que l'installation comprend 4 sous-systèmes, nous avons élaboré un modèle de référence pour le sous-système du bouilleur et un autre pour le condenseur et le receveur, ceci, afin de bien isoler les différents comportements possibles qui peuvent se produire dans chaque sous-système. Nous disposons de plusieurs ensembles de données historiques contenant aussi bien des situations de fonctionnement normal ainsi que des dysfonctionnements qui se sont produits dans le passé.

5.3.1 Bouilleur

5.3.1.1 Apprentissage hors ligne à partir des données brutes

Pour le sous-système du bouilleur, nous avons testé les performances de l'outil SALSA, dans un premier temps hors ligne, en réalisant un premier modèle de référence en ne prenant en compte que le régime en fonctionnement normal. Pour ceci, nous avons utilisé, comme descripteurs et valeurs pour la normalisation, la configuration de l'espace de représentation montrée dans le Tableau 5.7 Les descripteurs choisis ne correspondent qu'aux variables directement liées au fonctionnement du bouilleur et considérées par l'expert comme les plus représentatives. La Figure 5.14 illustre l'ensemble d'apprentissage (normalisé) utilisé pour la caractérisation des états fonctionnels du bouilleur en opération normale.

Tableau 5.7. Configuration de l'espace de description du bouilleur (données brutes)

Descripteur	Valeurs	
	Maximale	Minimale
F3	800	0
Q4	54	0
P7	8.5	5
L8	160	140
F10	20	0

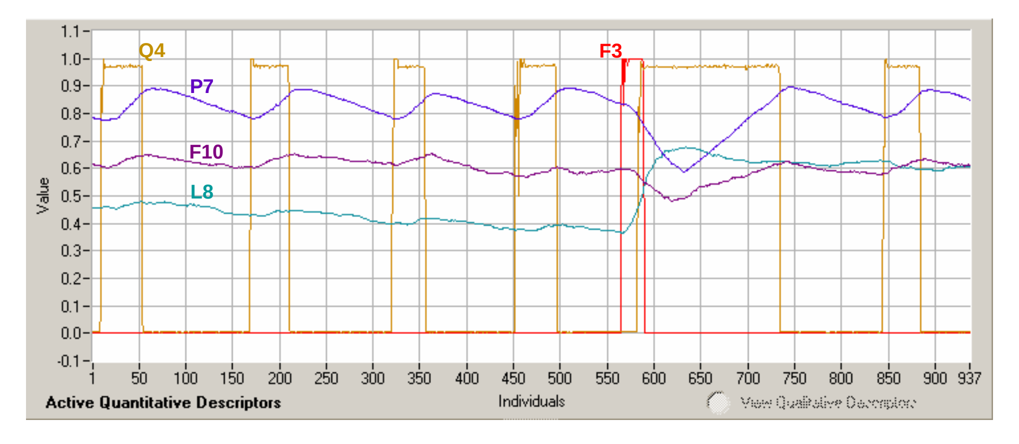
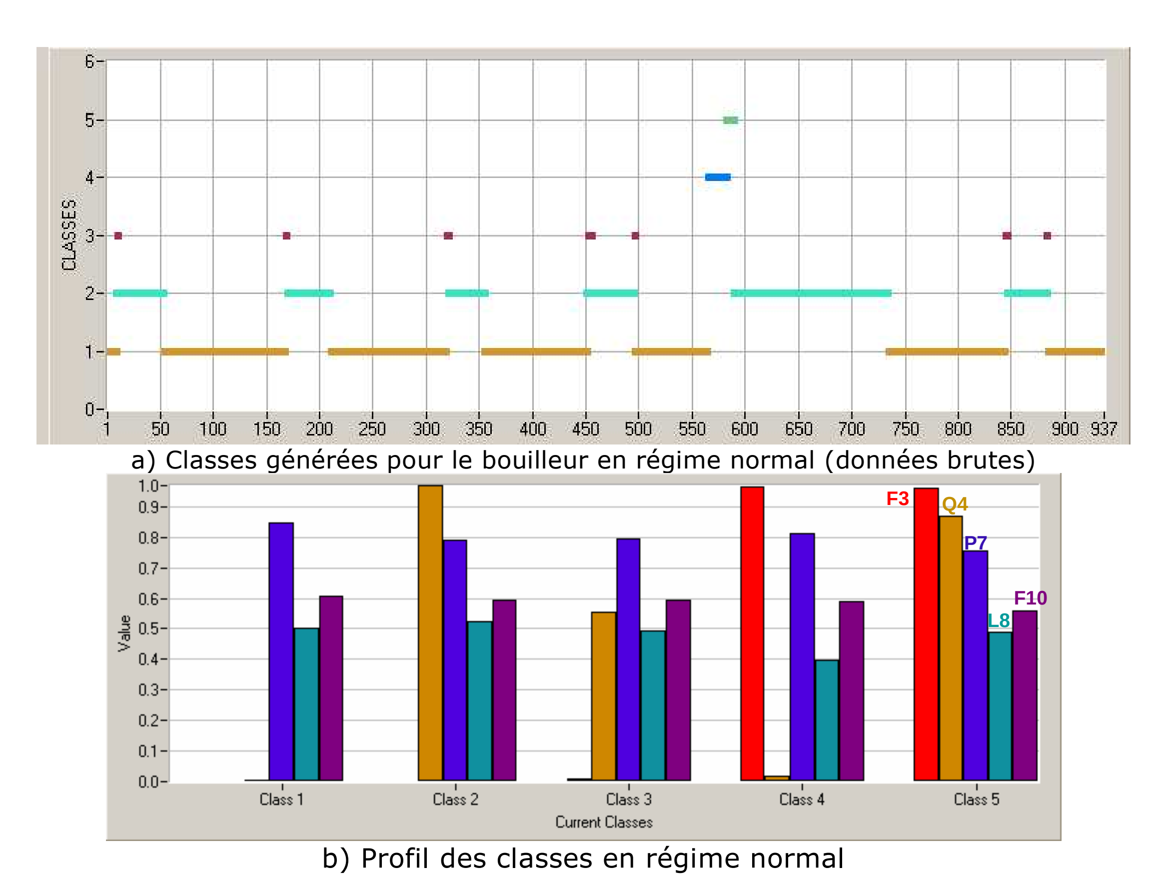


Figure 5.14. Ensemble d'apprentissage pour le bouilleur en opération normale

L'étape d'apprentissage, pour l'identification des états en régime normal du bouilleur, a été réalisée en choisissant un auto-apprentissage. La classification la plus représentative pour l'expert a été obtenue avec la fonction *Binomiale centrée* pour le calcul des DAM, la famille de connectifs flous *Minimum-Maximum*, avec une exigence $\alpha = 0.75$ pour l'adéquation globale. La Figure 5.15 illustre la classification résultante ainsi que le profil de chaque classe créée.



a) Classes générées pour le bouilleur en régime normal (données brutes)

b) Profil des classes en régime normal

Figure 5.15. Classification obtenue par l'auto-apprentissage pour le bouilleur

Les classes générées par SALSA ont été interprétées par l'expert à l'aide de leur profil. En régime d'opération normale, quatre situations peuvent se produire :

SITUATION	**Description**	**Classe**
Opération normale	Aucune régulation n'est nécessaire	*C1*
Régulation de la pression *P7*	Dès que la pression descend en dessous d'un seuil minimum la résistance se met à chauffer (*Q4 = 55kW*)	*C2 & C3*
Régulation du niveau *L8*	Si le niveau d'eau est inférieur à une valeur minimale la pompe P1 se met en marche et un débit d'eau en *F3* est observé.	*C4*
Régulation du niveau *L8* et de pression *P7*	Quand la température de l'eau provenant du receveur est très basse, la pression *P7* chute et donc *Q4* commence à chauffer avant que la pompe *P1* s'arrête.	*C5*

L'automate généré à partir de la séquence de classes obtenues est montré dans la Figure 5.16.

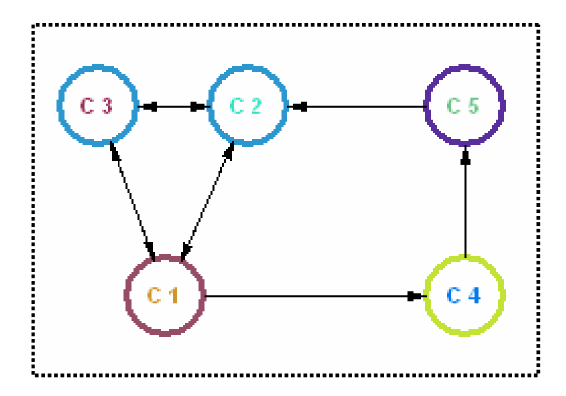
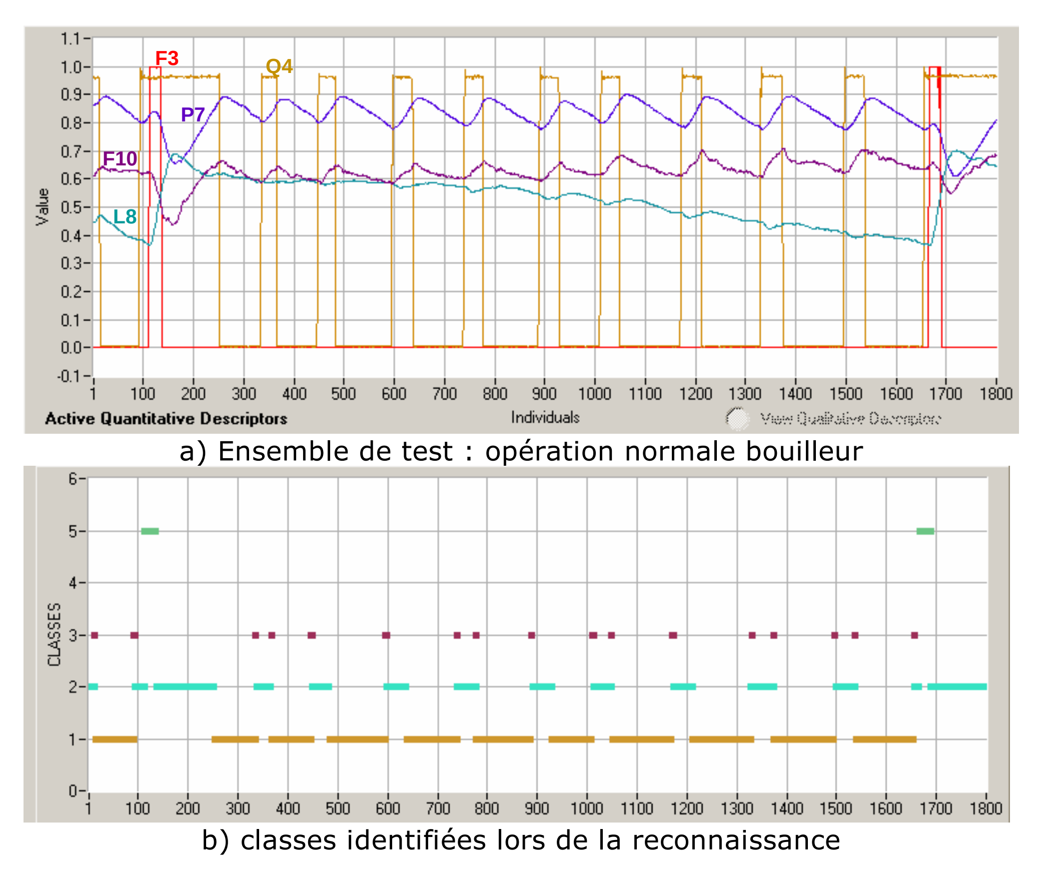


Figure 5.16. Automate pour le bouilleur en régime normal



a) Ensemble de test : opération normale bouilleur

b) classes identifiées lors de la reconnaissance

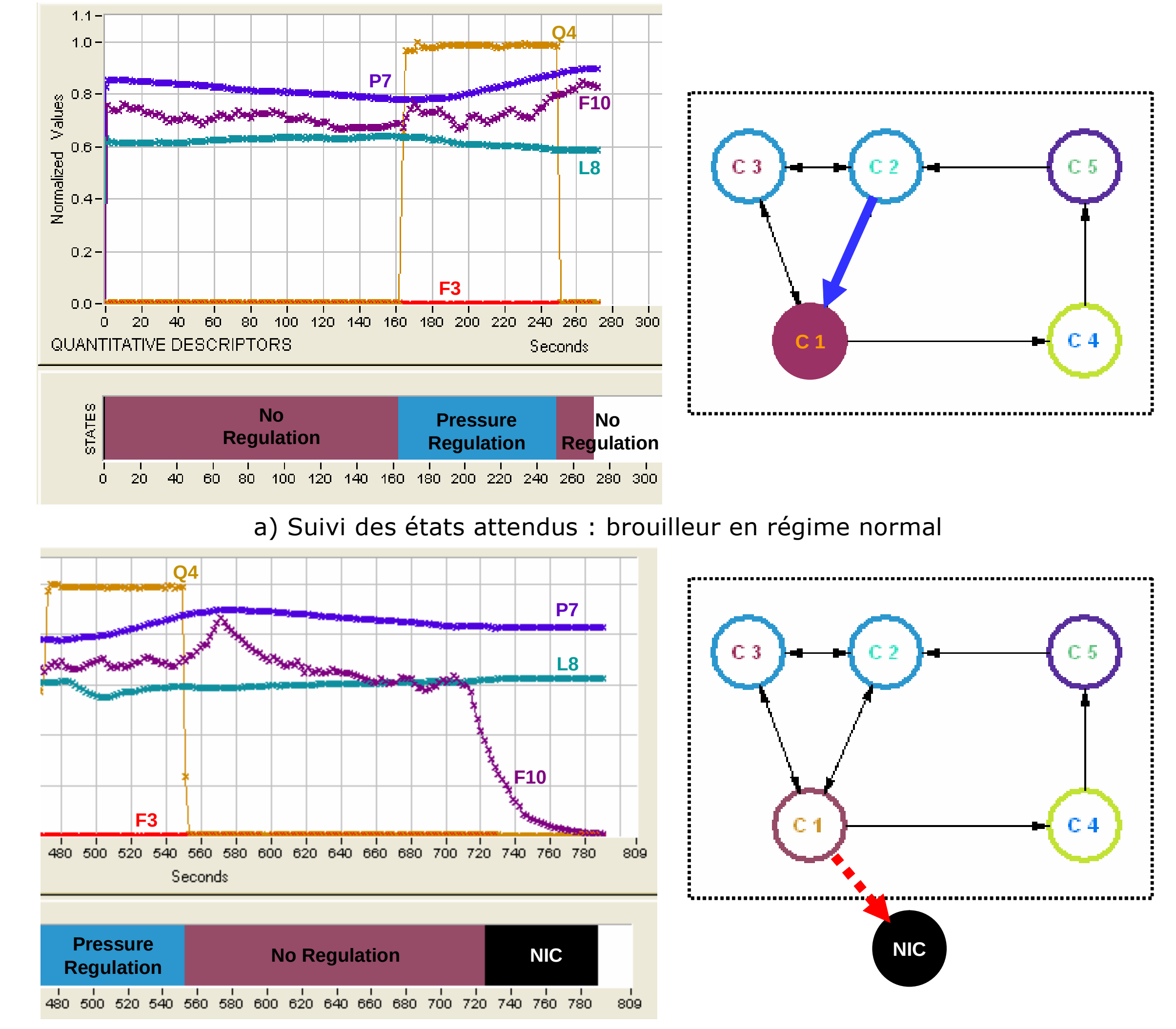
Figure 5.17. Validation de la classification obtenue pour l'opération normale du bouilleur

La classification obtenue par SALSA a été validée en utilisant un ensemble de données de test pour l'opération normale du brouilleur. La Figure 5.17 montre les résultats de la reconnaissance des données de test. Dans les données de test, seule une situation n'était pas présente (*C4- régulation du niveau d'eau)*, toutes les autres ont été correctement identifiées.

5.3.1.2 Identification en ligne

Pour la validation en ligne, nous avons utilisé le modèle de référence obtenu pour le sous-système du bouilleur en opération normale (voir Figure 5.16) et les mesures directes des capteurs et actionneurs de l'installation. Dans la Figure 5.18, nous observons les évolutions du brouilleur au fur et à mesure que des observations se

présentent. La Figure 5.18a) montre la reconnaissance d'un état attendu. Ici, le brouilleur bascule de la régulation de pression *P7* (*C2*) vers l'état où la régulation n'est pas active (*C1*). En revanche, la détection d'une déviation est représentée par le "rejet" (non reconnaissance) des observations qui fait basculer l'automate d'un état attendu (*C1*) vers la *NIC*, comme le montre la Figure 5.18b).



a) Suivi des états attendus : brouilleur en régime normal

b) Détection d'une déviation : observations non reconnues

Figure 5.18. Identification en ligne des états du brouilleur

Dès lors qu'une déviation est détectée, nous procédons à sa caractérisation par le biais d'un *apprentissage supervisé actif*. Pour ceci, nous gardons les mêmes paramètres de l'algorithme LAMDA et nous autorisons la création des nouvelles classes à partir des observations non reconnues en ayant comme classes prédéfinies celles obtenues précédemment lors de l'auto-apprentissage. Deux nouvelles classes ont été alors créées (voir Figure 5.19). Une de ces nouvelles classes a été caractérisée comme une situation normale (non présente lors de la phase d'apprentissage), correspondant à une hausse du débit de vapeur à la sortie du bouilleur (*C6*). La deuxième classe correspond à une situation de défaillance : il s'agit d'un blocage à la sortie du bouilleur (*C7*).

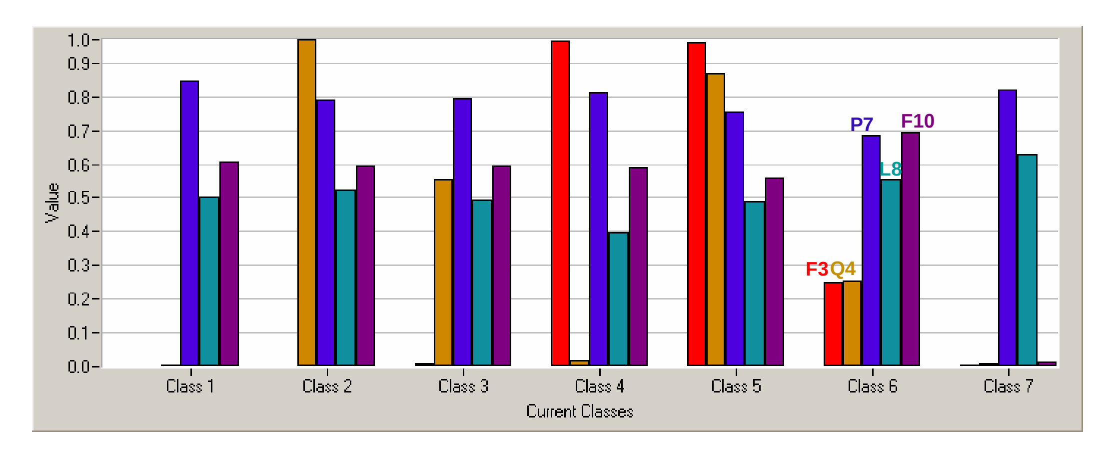


Figure 5.19. Profil des classes après un apprentissage supervisé actif

Les nouvelles situations sont ajoutées au modèle de référence. Le nouvel automate est illustré dans la Figure 5.20

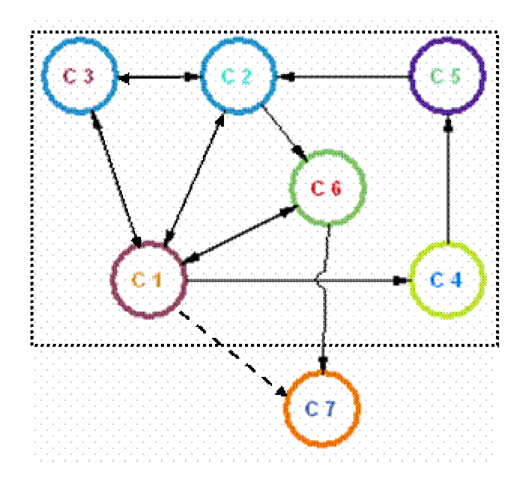


Figure 5.20. Modèle de référence pour le bouilleur

Dans le modèle de la Figure 5.20, l'expert a ajouté une transition de la classe *C1* vers la classe *C7*, dans la mesure où la défaillance peut se produire directement après une situation de fonctionnement normal.

5.3.1.3 Apprentissage hors ligne à partir des épisodes

Les types d'épisodes obtenus des variables utilisées pour l'identification et la caractérisation des états du générateur de vapeur sont présentés dan le Tableau 5.8.

Tableau 5.8 Types d'épisodes pour les variables du générateur de vapeur

	A : descente rapide		L : montée rapide		G : stable
	S : descente		R : montée		O : stable bas
	F : descente lente		H : montée lente		P : stable haut
low	Lo : bas (0)	high	Hi : Haut (1)	?	Z : inconnu

En ce qui concerne le sous-système du bouilleur, l'objectif était de caractériser son fonctionnement en régime normal ainsi que les situations de défaillances connues. Il s'agit de réaliser un apprentissage dirigé. Donc, à partir des informations issues de la représentation des variables à base d'épisodes, les éléments de l'ensemble d'apprentissage ont été étiquetés de manière à représenter au mieux chaque situation.

Pour l'identification de ces situations, 4 variables ont été sélectionnées pour construire l'espace de description : le débit de la vapeur à la sortie (F10), le débit d'eau fourni par la pompe (F3), la pression à l'intérieur du bouilleur (P7) et le niveau d'eau (L8). Pour chaque variable, trois informations concernant les épisodes ont été choisies : l'épisode courant (*ActEp*), l'épisode précédent (*PreEp*) et la valeur finale de l'épisode précédent (*Fval*). Les deux premiers descripteurs sont de type qualitatif et le troisième de type quantitatif. Pour P7 et L8 une information supplémentaire a été prise en compte; il s'agit de la pente de l'épisode précédent. Le Tableau 5.9 montre l'espace de description choisi pour le bouilleur.

Tableau 5.9. Espace de description à partir d'épisodes pour le bouilleur

DESCRIPTEURS	TYPE	VALEURS	
D1. F10,ActEp	Qualitatif	Modalités : a, f, g, h, l, o, p, r, s	
D2. F10,PreEp	Qualitatif	Modalités : a, f, g, h, l, o, p, r, s	
D3. F10,Fval	Quantitatif	Min =0	Max =20
D4. F3,ActEp	Qualitatif	Modalités : lo, hi	
D5. F3,PreEp	Qualitatif	Modalités : lo hi	
D6. F3,Fval	Quantitatif	Min =0	Max =800
D7. P7,ActEp	Qualitatif	Modalités : a, f, g, h, l, o, p, r, s, low	
D8. P7,PreEp	Qualitatif	Modalités : a, f, g, h, l, o, p, r, s, low	
D9. P7,Fval	Quantitatif	Min =6	Max =8.3
D10. P7,Slope	Quantitatif	Min = -0.229	Max = 8.3
D11. L8,ActEp	Qualitatif	Modalités : a, f, g, h, l, o, p, r, s	
D12. L8,PreEp	Qualitatif	Modalités : a, f, g, h, l, o, p, r, s	
D13. L8,Fval	Quantitatif	Min =145	Max =157
D14. L8,Slope	Quantitatif	Min =-2.272	Max =2.587

Les situations qui devaient être identifiées sont décrites dans le Tableau 5.10. Les épisodes type caractérisant chaque situation y sont présentés.

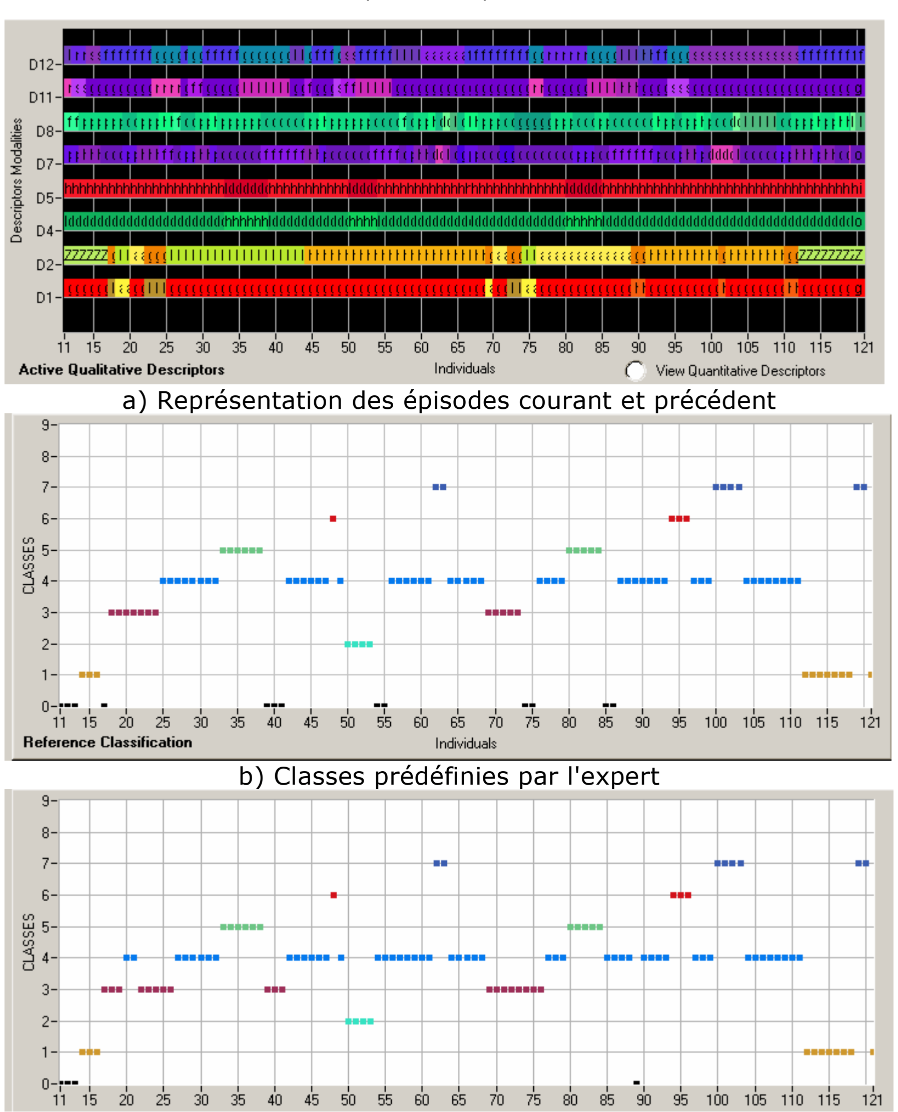
L'ensemble d'apprentissage contient plusieurs fois les différentes situations et les éléments ne sont pas tous étiquetés.

Nous avons testé plusieurs configurations de l'algorithme LAMDA pour apprendre ces différentes situations. Pour la validation du classificateur, nous avons fait la reconnaissance de l'ensemble d'apprentissage afin de vérifier si l'outil été capable d'identifier correctement les situations apprises. Dans les Figure 5.21 et Figure 5.22, nous illustrons un aperçu des résultats obtenus par deux classificateurs.

Tableau 5.10. Différentes situations du bouilleur à être identifiées

SITUATION	CLASSE	EPISODES - VALEURS
Opération normale : (démarrage) Aucune régulation - Peu de vapeur en sortie	*C1*	F10,ActEp: g F10,PasEp: Z F10,Val: ~0
Opération normale : Régulation niveau (*L8*) d'eau	*C2*	F3,Act: hi L8,Val: ~147
Opération normale : Aucune régulation	*C4*	F10,ActEp: g P7,ActEp: p, h, o L8,ActEp: g
Blocage de la vapeur à la sortie du bouilleur	*C3*	F10,ActEp: a, g, l F10,Val: < 3
Défaillance de la pompe : se met en marche	*C5*	F3,ActEp: hi L8,Val: 149 à 153
Fuite d'eau du bouilleur	*C6*	L8,ActEp: s, a L8,Val: 150 à 147.8
Défaillance du capteur de pression P7	*C7*	P7,ActEp: low, l P7,Fval: <<1

- Classificateur 1 : *Binomiale centrée, Produit,* $\alpha = 1$



a) Représentation des épisodes courant et précédent

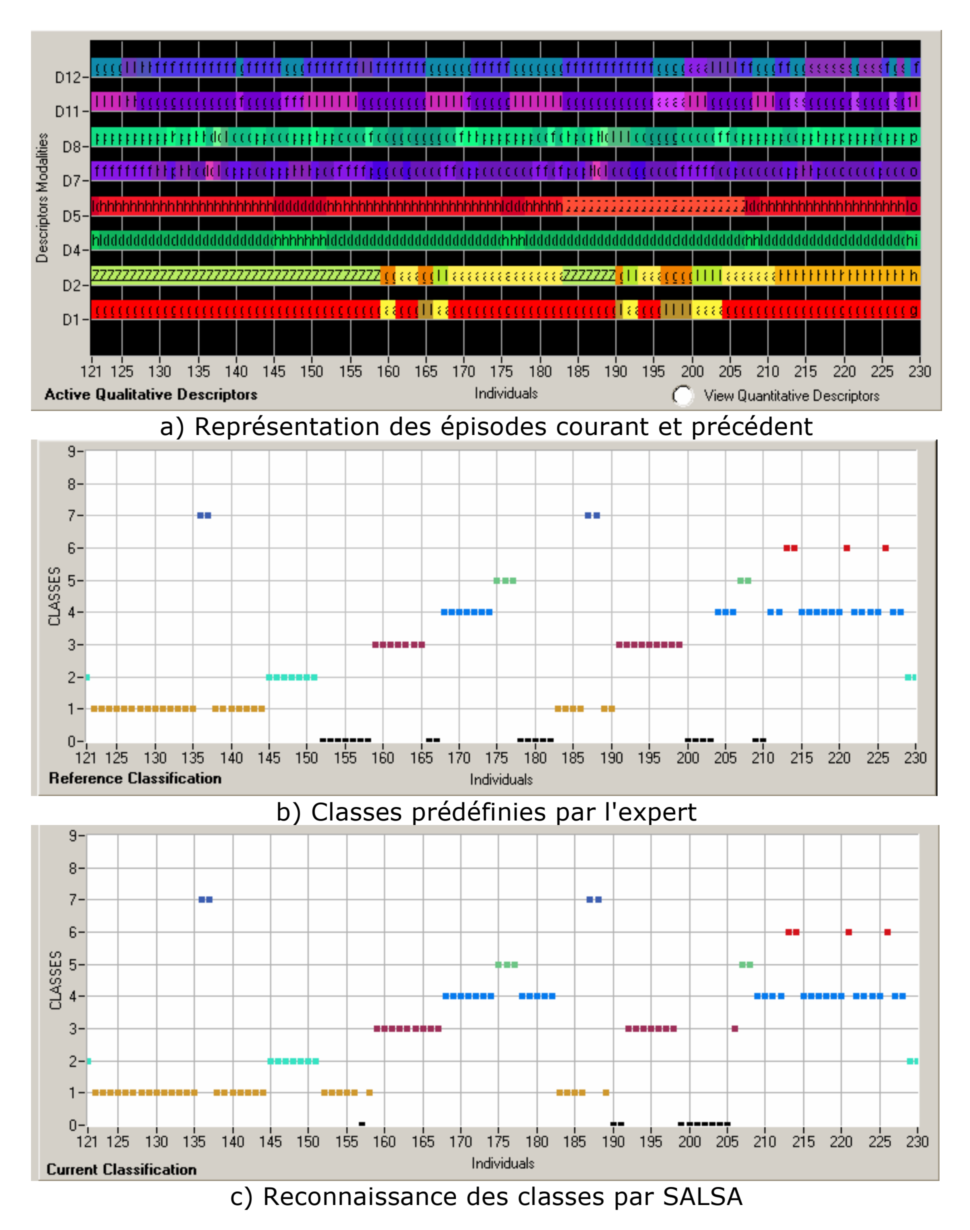
b) Classes prédéfinies par l'expert

c) Reconnaissance des classes par SALSA

Figure 5.21. Résultats de la reconnaissance pour le classificateur 1

Avec cette sélection des paramètres, et parmi les 312 éléments étiquetés par l'expert, un seul élément n'a pas été reconnu (0.32%), et 15 éléments (4.82%) ont été mal reconnus.

- <u>Classificateur 2 :</u> *Gaussienne, Produit,* $\alpha = 0.99$



a) Représentation des épisodes courant et précédent

b) Classes prédéfinies par l'expert

c) Reconnaissance des classes par SALSA

Figure 5.22. Résultats de la reconnaissance pour le classificateur 2

En utilisant les paramètres de classification ci-dessus, plus d'éléments (21 soit 6.38%) ont été "rejetés" mais seulement 3 (0.97%) ont été mal classés.

La reconnaissance en ligne a été effectuée pour les deux classificateurs. Nous avons constaté que, pour certaines situations de défaillances telle que la fuite du bouilleur, il n'y a pas suffisamment d'éléments pour réaliser un bon apprentissage et, donc, lors de la reconnaissance en ligne, la détection de fausses alarmes a lieu. Une situation parfois difficile à identifier, tant avec des données brutes qu'avec des

épisodes, est le blocage de la vapeur à la sortie du bouilleur quand le débit de vapeur est très faible avant que le blocage ne se produise.

5.3.2 Condenseur et receveur

Pour les sous-systèmes du condenseur et du receveur, nous présentons les performances de l'outil SALSA à partir des épisodes. L'utilisation des données brutes ne nous a permis de caractériser de façon satisfaisante les différents états dans lesquels ce sous-système peut se trouver.

5.3.2.1 Apprentissage hors ligne à partir des épisodes

De manière analogue que pour le bouilleur, les différentes situations pour les sous-systèmes du condenseur et du receveur, devaient être apprises à partir d'un ensemble d'apprentissage dont les éléments étaient déjà étiquetés par l'expert. Les situation à identifier sont : l'opération en régime normal, une fuite au niveau du condenseur et une fuite au niveau du receveur. Pour ce faire, trois variables ont été utilisées pour l'élaboration de l'espace de représentation : la position de la vanne de décharge (*V5comm*), le niveau d'eau dans le condenseur (L8) et le niveau d'eau du receveur (L1). Pour chaque variable, nous avons sélectionné 3 informations concernant les épisodes : l'épisode courant (*ActEp*), l'épisode précédent (*PreEp*) et la valeur finale de l'épisode précédent (*Fva*l). Les deux premiers descripteurs sont de type qualitatif et le troisième de type quantitatif. Pour L18 et L1, une information supplémentaire a été prise en compte : il s'agit de la pente de l'épisode précédent (*Slope*). Le Tableau 5.11 montre l'espace de description choisi pour le condenseur et le receveur.

Tableau 5.11. Descripteurs pour le condenseur et le receveur

DESCRIPTEUR	TYPE	VALEURS	
D1. V5com,ActEp	Qualitatif	Modalités : lo, hi	
D2. V5com,PreEp	Qualitatif	Modalités : lo, hi	
D3. V5com,Fval	Quantitatif	Min = 0	Max = 1
D4. L18,ActEp	Qualitatif	Modalités: a, f, g, h, l, o, p, r, s, low	
D5. L18,PreEp	Qualitatif	Modalités: a, f, g, h, l, o, p, r, s, low	
D6. L18,Fval	Quantitatif	Min = 4	Max = 7
D7. L18,Slope	Quantitatif	Min = -0.045	Max = 0.03
D8. L1,ActEp	Qualitatif	Modalités : a, f, g, h, l, o, p, r, s	
D9. L1,PreEp	Qualitatif	Modalités : a, f, g, h, l, o, p, r, s	
D10. L1,Fval	Quantitatif	Min = 300	Max = 360
D11. L1,Slope	Quantitatif	Min = -0.7	Max = 3.2

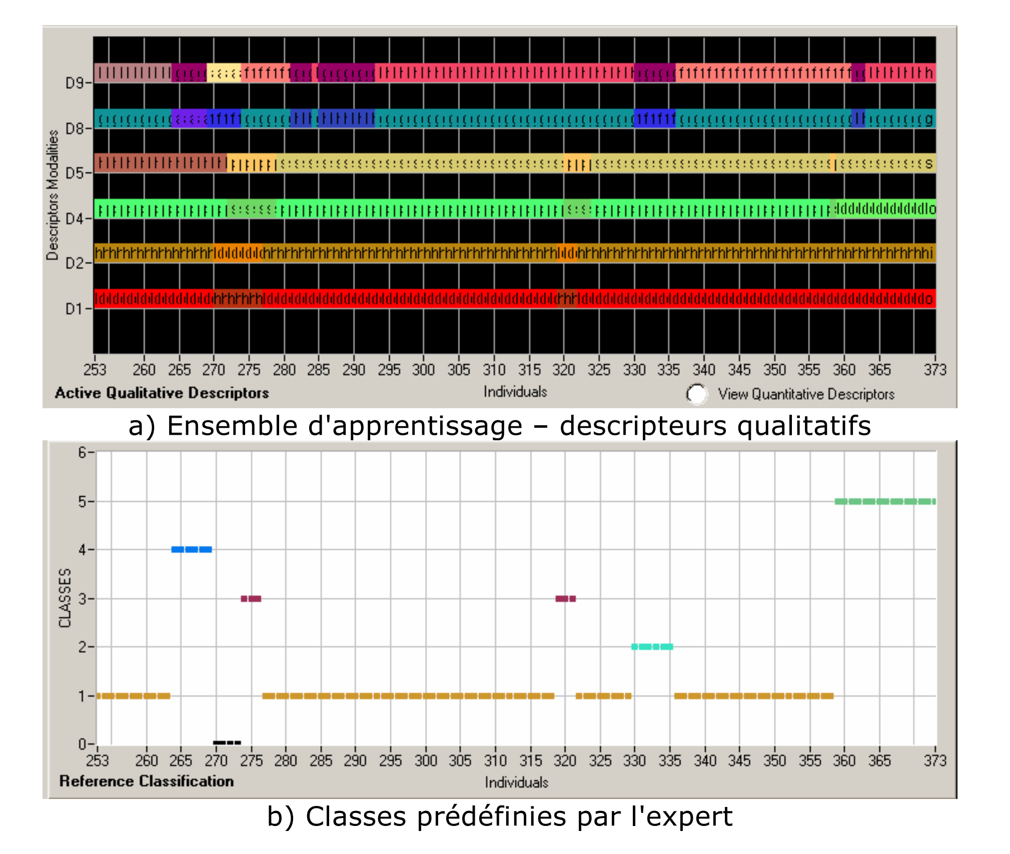
A partir des informations issues des épisodes, l'étiquetage des éléments a été fait en considérant le descriptif du Tableau 5.12. La fuite du receveur est caractérisée par une décente rapide tandis que la diminution du niveau du receveur, due au

démarrage de la pompe pour réguler le niveau du bouilleur, est représentée par une décente lente.

Tableau 5.12. Situations du condenseur et du receveur à identifier par SALSA

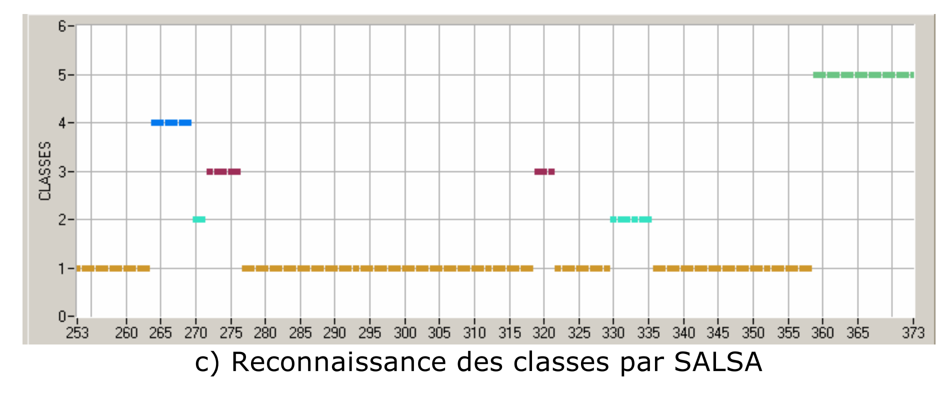
SITUATION	CLASSE	EPISODES - VALEURS
Opération normale : Aucune régulation	*C1*	V5com,ActEp: lo L18,ActEp: p L1: g
Opération normale : Pompe P1 allumée – régulation L8	*C2*	L1,ActEp: f
Opération normale : V5 ouverte – régulation L18	*C3*	V5com,ActEp: hi L18,ActEp: s L18,PreEp: p
Fuite du receveur	*C4*	L1,ActEp: a, f
Fuite condenseur	*C5*	V5com,ActEp: lo L18,ActEp: lo

Les paramètres de l'algorithme LAMDA qui ont été utilisés pour la validation de la caractérisation des situations à partir de l'apprentissage dirigé sont : la fonction *gaussienne* pour le calcul des adéquations marginales, le *produit* pour l'agrégation des contributions des DAMs et une exigence de 0.9. Un aperçu des résultats de la reconnaissance des éléments de l'ensemble d'apprentissage est illustré dans la Figure 5.23. Tous les éléments étiquetés ont été classés, néanmoins 19 (4.59%) d'entre eux ont été mal reconnus. Cependant, les défaillances apprises ont été correctement identifiées.



a) Ensemble d'apprentissage – descripteurs qualitatifs

b) Classes prédéfinies par l'expert



c) Reconnaissance des classes par SALSA

Figure 5.23. Résultats de l'identification des états pour le condenseur et le receveur

Le profil des classes généré à partir des paramètres de LAMDA est illustré dans la Figure 5.24. Les descripteurs suivent l'ordre donné dans le Tableau 5.11.

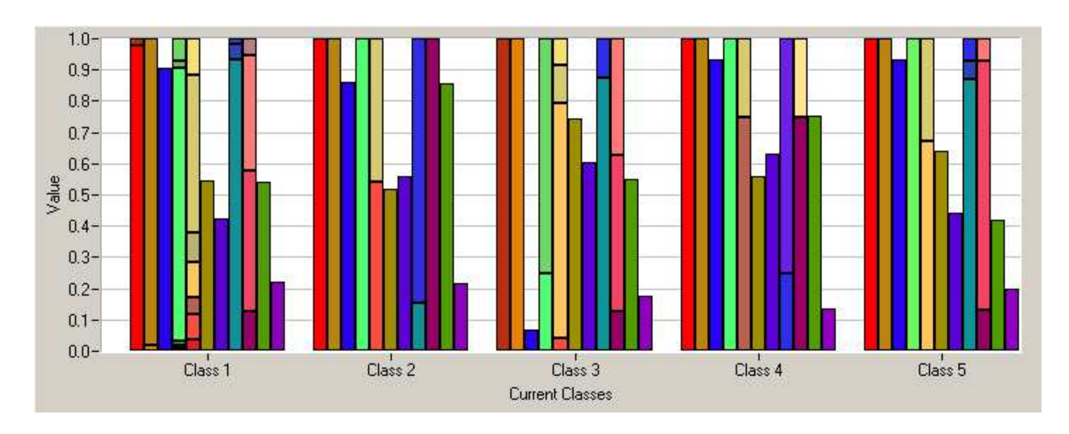


Figure 5.24. Profil des classes pour le condenseur et le receveur

L'automate résultant pour les états du condenseur et du receveur est présenté dans la Figure 5.25.

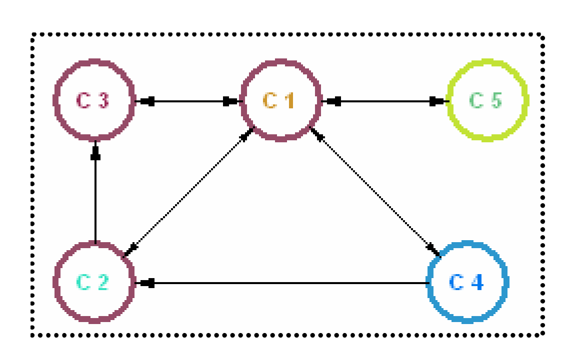


Figure 5.25. Automate issu de l'apprentissage séquentiel pour le condenseur et le receveur.

5.4 Haut Fourneau pour la production de fer (Redcar (Corus) Blast Furnace)

Dans le haut fourneau, l'oxyde de fer est **réduit** pour produire du fer métallique. Le fer est ensuite fondu à une température de 1500°C. Les déchets liquides sont aussi enlevés. Ces déchets sont formés de composants non-ferriques présents dans le minerai (principalement chaux, silice, magnésium, aluminium).

L'air chaud (1200 °C) insufflé à la base du fourneau provoque la combustion du coke (carbone presque pur). L'oxyde de carbone ainsi formé va "réduire" les oxydes de fer, c'est-à-dire leur prendre leur oxygène et, de ce fait, isoler le fer du minerai.

oxyde de fer(III)	+	*monoxyde de carbone*	➡	*fer*	+	*dioxyde de carbone*
$Fe_2O_3(s)$	**+**	**$3\ CO(g)$**	**➡**	**$2\ Fe(s)$**	**+**	**$3\ CO_2(g)$**

La température du fourneau étant de l'ordre de 1500°C, la chaleur dégagée par la combustion fait fondre le fer et la gangue. La gangue, de densité moindre, flotte sur un mélange à base de fer appelé "***fonte***". La fonte est recueillie après que le fer, plus dense, soit évacué en fond de fourneau.

La production de fer en haut fourneau est un processus continu. Le fourneau est chauffé constamment et est re-chargé avec des matières premières par le haut pendant que le fond du fourneau est soutiré (voir Figure 5.26). La production de fer peut se faire en continu durant dix ans avant de renouveler la couche interne de protection du fourneau.

Le haut fourneau lui même est un récipient refroidi à l'eau, de coupe transversale circulaire, de 30m d'hauteur. Des couches de coke et de minerai sont alternativement chargées par le haut, de façon contrôlée. L'air à environ 1200°C est insufflé par le fond du fourneau, au travers d'une tuyère de cuivre refroidie à l'eau. L'air réagit avec le coke et du fuel injecté en supplément, pour générer la chaleur nécessaire pour faire fondre le fer et la gangue ainsi que pour former un gaz "réducteur" de CO et H2. Le minerai a besoin de 7 heures pour être "réduit" et fondre au fur et à mesure qu'il descend au travers des différentes piles. Le temps de séjour pour les gaz qui remontent est de l'ordre de quelques secondes.

Le fourneau est toujours rempli en chargeant un nouveau lot de minerai dès qu'une chute de niveau est détectée dans la partie supérieure du fourneau. La descente du minerai est généralement stable à 5 mètres/heure. Le fer liquide et la gangue recueillis dans l'âtre, construit avec des briques réfractaires, sont enlevés périodiquement par éclusage.

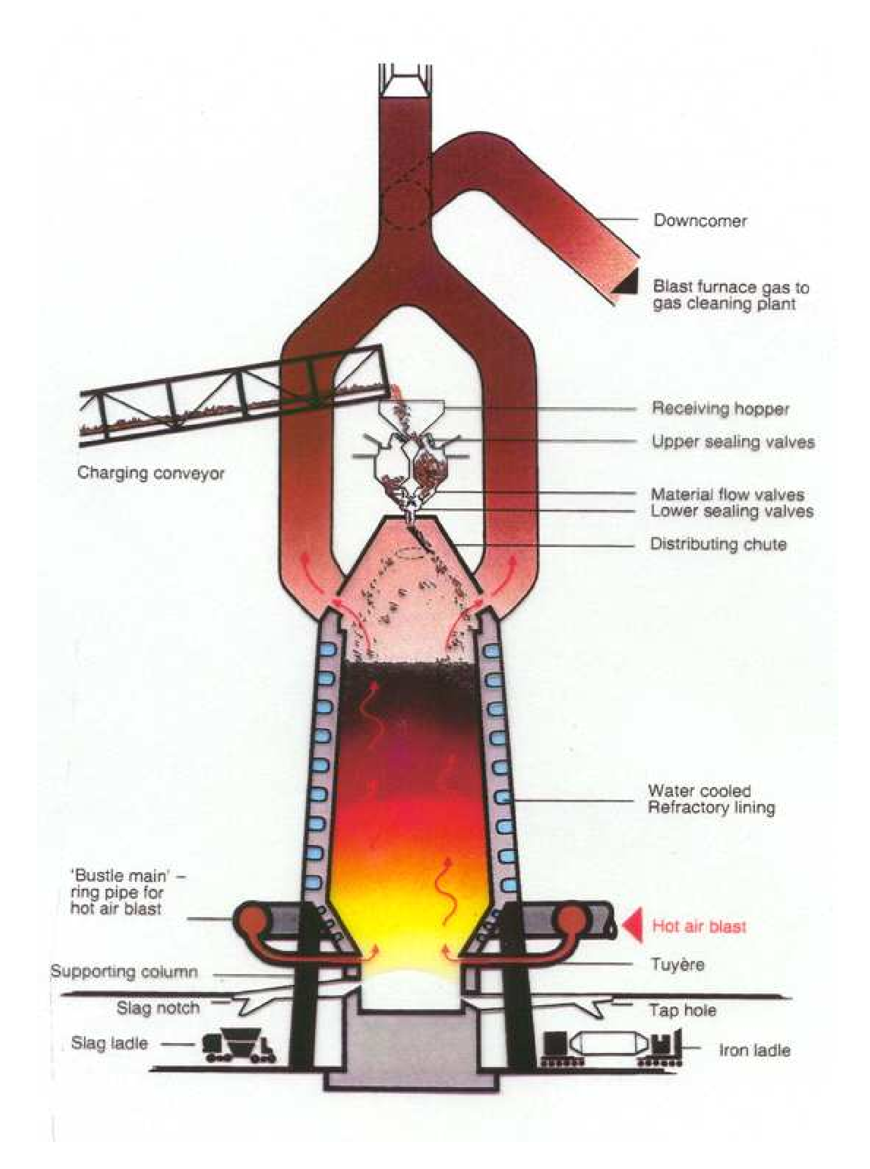


Figure 5.26. Schéma d'un haut fourneau

Le haut fourneau de Redcar est le plus grand du Royaume Uni. Il produit 64000 tonnes de fer liquide par semaine. Chaque heure, 5 lots de minerai de 120 tonnes chacun sont chargés avec la même quantité de lots de coke, chacun de 30 tonnes. La chaleur est enlevée des parois à travers de larges panneaux refroidis à l'eau, sur 13 rangées. Les rangées, 7 à 10, sont faites de cuivre, et celles d'en-dessous et d'au-dessus sont faites de "fonte". L'installation de Redcar contient 4 colonnes de pression. La pression est mesurée entre les rangées 6 et 13, pour chaque colonne de pression (Δp).

Sous opération en régime normal, la pression doit diminuer graduellement entre le fond (rangée 6) et le haut (rangée 11) de la section du fourneau examinée. Les opérateurs se sont aperçus que les tendances des pressions différentielles (*chutes de pression*) peuvent être un bon indicateur de l'instabilité du fourneau.

Un événement **instable** se produit quand la matière dans la pile du fourneau "*glisse*" soudainement au lieu de descendre graduellement.

L'objectif était donc d'analyser les tendances des pressions différentielles des rangées 6 à 9 de manière à déterminer s'il est possible de les utiliser pour **prédire qu'un "glissement" s'est produit**. Les opérateurs ont observé qu'avant qu'un glissement se produise, plus d'une mesure de Δp doit avoir des tendances similaires.

Pour réaliser ceci, nous avons choisi, à l'aide de l'expert, des ensembles de données historiques contenant des éléments fortement liés à des états de pré-faute (***prédiction*** *d'un glissement)* et de faute (***détection*** *d'un glissement)*.

Nous présentons, par la suite, les résultats de l'apprentissage hors ligne à partir d'épisodes.

5.4.1 Apprentissage hors ligne à partir des épisodes

Les variables brutes qui ont été utilisées pour la représentation en épisodes étaient les différentiels de pression entre les rangées 6 et 9 de chacune des colonnes (Δp) du fourneau avec une période d'échantillonnage de 1 élément/min (Figure 5.27).

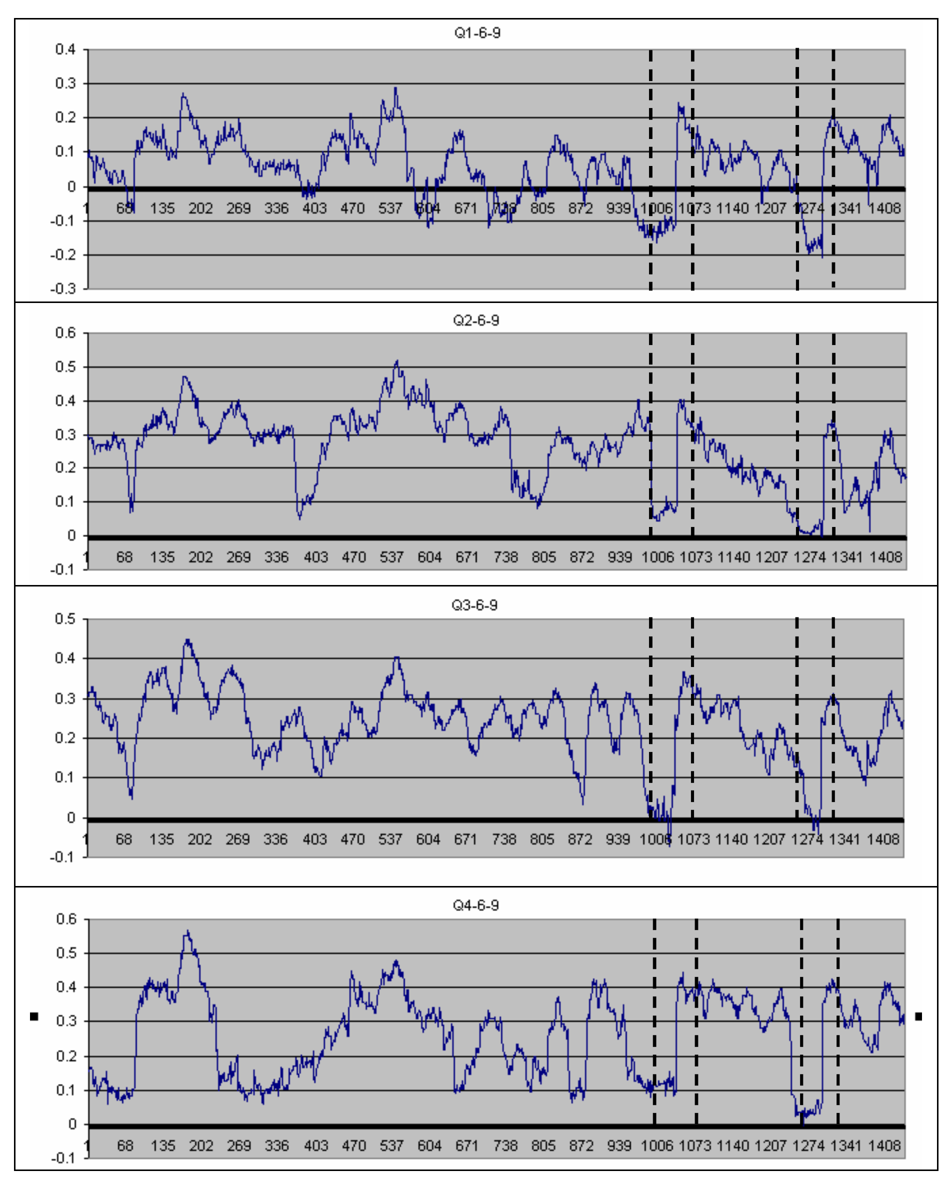
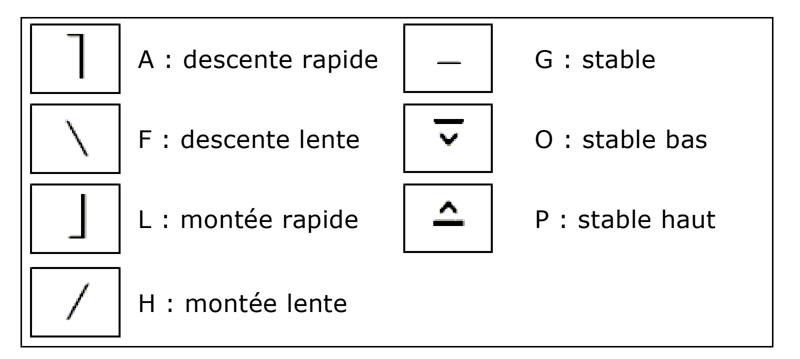


Figure 5.27. Données brutes pour le haut fourneau

Sur les données brutes de la Figure 5.27, l'expert a localisé deux conditions d'instabilité (glissements), délimitées par les traits pointillés. Cinq types d'épisodes ont été obtenus pour représenter les tendances des pressions (*Q1, Q2, Q3, Q4*) du fourneau. Ces épisodes sont illustrés dans le Tableau 5.13.

Tableau 5.13. Episodes représentatifs des variables du haut fourneau



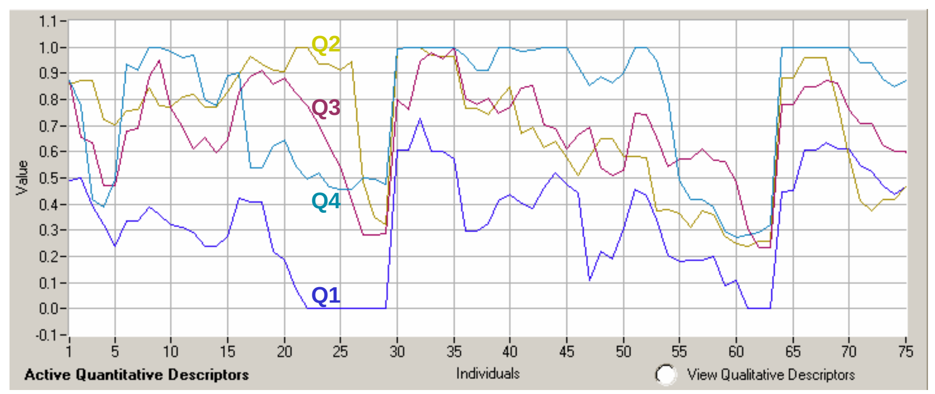
	A : descente rapide		G : stable
	F : descente lente		O : stable bas
	L : montée rapide		P : stable haut
	H : montée lente		

Les descripteurs utilisés pour l'apprentissage hors ligne sont des informations qualitatives issues de la représentation à partir des épisodes (épisode courant *ActEp,* épisode précédent *PasEp*) et la valeur de chaque différentiel de pression (*Val*) à chaque instant qu'un épisode s'est produit. La configuration des descripteurs est donnée dans le Tableau 5.14. L'ensemble des données utilisé pour l'apprentissage est présenté dans la Figure 5.28. Dans cet ensemble, deux conditions d'instabilité sont présentes (individus 25 à 32, et individus 60 à 65).

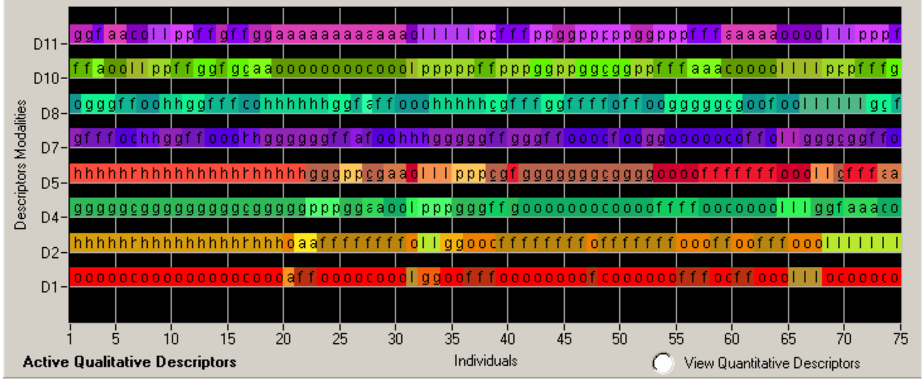
Tableau 5.14. Configuration des descripteurs du haut fourneau

DESCRIPTEUR	TYPE	VALEURS	
D1. Q1,ActEp	Qualitatif	Modalités : a, f, l, h, g, o, p	
D2. Q1,PreEp	Qualitatif	Modalités : a, f, l, h, g, o, p	
D3. Q1,Val	Quantitatif	Min = -0.1	Max = 0.35
D4. Q2,ActEp	Qualitatif	Modalités: a, f, l, h, g, o, p	
D5. Q2,PreEp	Qualitatif	Modalités: a, f, l, h, g, o, p	
D6. Q2,Val	Quantitatif	Min = -0.1	Max = 0.35
D7. Q3,ActEp	Qualitatif	Modalités : a, f, l, h, g, o, p	
D8. Q3,PreEp	Qualitatif	Modalités : a, f, l, h, g, o, p	
D9. Q3,Val	Quantitatif	Min = -0.1	Max = 0.35
D10. Q4,ActEp	Qualitatif	Modalités : a, f, l, h, g, o, p	
D11. Q4,PreEp	Qualitatif	Modalités : a, f, l, h, g, o, p	
D12. Q4,Val	Quantitatif	Min = -0.1	Max = 0.35

Les experts n'étaient intéressés que par l'identification des situations de pré-faute et de faute; un auto-apprentissage a été réalisé. La classification présentée dans la Figure 5.29 a été obtenue en utilisant la fonction *gaussienne* pour le calcul des adéquations marginales (DAMs), le *Minimum* comme connectif avec une exigence $\alpha = 0.765$.



a) Descripteurs quantitatifs – valeurs des différentiels de pression



b) Descripteurs qualitatifs – épisode courant et épisode précédent

Figure 5.28. Espace de représentation pour le haut fourneau

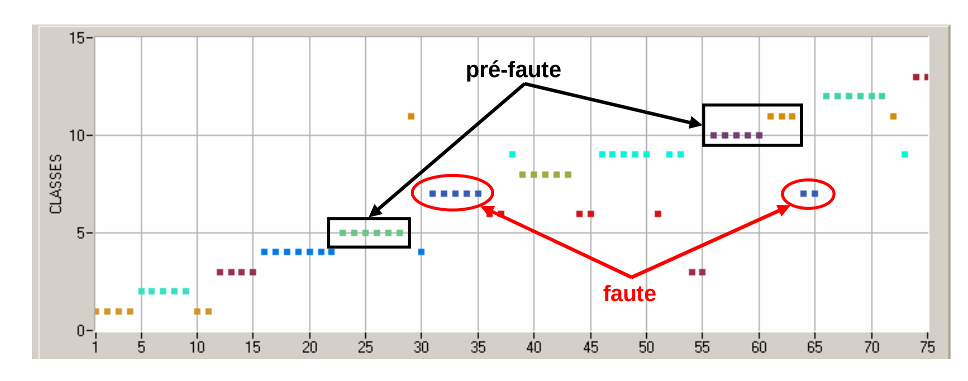


Figure 5.29. Résultats de l'auto-apprentissage pour la prédiction et la détection des glissements

La *pré-faute* est représentée par la classe 5 (la première occurrence) et par les classes 10 et 11 (deuxième occurrence). En ce qui concerne la situation de *faute,* elle est représentée par la classe 7. La caractérisation de la pré-faute et de la faute est montrée dans le Tableau 5.15.

Tableau 5.15. Caractérisation des situations de pré-faute et faute pour le haut fourneau

SITUATION	VARIABLE	SEQUENCE D'EPISODES
Pré-faute	Q1	O, A, F
C5, C10, C11	Q2	G, A
	Q3	G, F, A
	Q4	G, A
faute	Q1	O, L
C7	Q2	O, L
	Q3	O, L
	Q4	O, L

5.5 Conclusion

Dans ce chapitre, nous avons présenté les résultats obtenus lors de l'utilisation de l'outil SALSA sur différentes unités pilotes. Pour certaines applications l'identification des situations n'était pas possible en utilisant uniquement les données brutes et une représentation à partir d'épisodes a été effectuée.

Nous avons, aussi, abordé les différentes étapes de la stratégie que nous proposons pour l'identification des états fonctionnels du générateur de vapeur. Dans un premier temps, nous avons réalisé un auto-apprentissage hors ligne pour caractériser les situations en fonctionnement normal. Puis, la reconnaissance en ligne des situations apprises a été réalisée. Nous avons illustré, aussi, la caractérisation des nouvelles situations lors de la détection d'une déviation.

CONCLUSION ET PERSPECTIVES

Dans notre travail nous avons proposé une stratégie d'aide à la décision pour la surveillance en ligne des processus complexes. Ces travaux ont été basés sur l'exploitation des données acquises sur le système surveillé. Cette stratégie suppose deux étapes. La première consiste en l'élaboration d'un modèle de référence des états normaux ou anormaux, à partir d'un apprentissage hors ligne. Cet apprentissage utilise des données historiques et, grâce à un dialogue avec les experts et les opérateurs, un ajustement des paramètres du classificateur pour fournir la meilleure représentation est possible. La deuxième étape, en prise directe avec les mesures accessibles en ligne, réalise la reconnaissance des états de fonctionnement connus. Cette étape d'identification en ligne, doit fournir l'image la plus informative et la mieux compréhensible par les opérateurs « in situ ».

Nous nous sommes basés sur la méthode de classification LAMDA, laquelle repose sur le concept de degré d'appartenance (adéquation) d'un objet aux classes existantes, qui remplace les critères classiques de distance. Cette méthode se différencie d'autres techniques par la notation de l'adéquation nulle et l'incidence de celle-ci sur la classification. La méthode comporte deux opérations principales : le calcul du degré d'adéquation marginal (DAM) et le calcul du degré d'adéquation global (DAG). Le DAM est calculé à partir d'un choix parmi plusieurs fonctions d'appartenance ayant des propriétés spécifiques, et à l'aide d'une fonction d'adéquation paramétrable. La sélectivité de l'algorithme peut être modulée par le paramètre "d'exigence" de cette fonction.

La représentation des fonctionnements à l'aide de classes a été complétée par la construction d'une machine à états finis ou automate, à partir de l'estimation des fonctions de transition. Cet automate peut être utilisé pour le suivi des états fonctionnels connus, ainsi que pour la détection des défaillances. A partir de l'élaboration de cet automate, deux types de défaillance peuvent être traités : des défaillances au niveau d'états, au travers de la non reconnaissance des observations en ligne, et des défaillances au niveau des transitions, au travers des passages non valides entre états. De plus, le fait de présenter à l'expert les descripteurs qui ont eu un

changement important dans les transitions observées, est une aide supplémentaire pour celui-ci dans un contexte de diagnostic.

Enfin, nous avons montré le mode opératoire et la faisabilité de notre proposition sur divers exemples issus de systèmes réels. Deux applications industrielles en vraie grandeur ont pu être abordées. Ces travaux ont été en partie réalisés dans le cadre du projet CHEM. Le but initial de ce projet était lié à la demande des industriels chimiques et pétrochimiques concernant la disponibilité d'outils d'aide à la décision dans le suivi de diverses opérations propres aux procédés complexes, et leurs possibles défaillances.

Suite à notre travail, plusieurs perspectives peuvent être envisagées notamment au niveau de la méthode de classification choisie pour le développement de la stratégie de surveillance que nous avons proposée.

En effet, le choix de LAMDA comme méthode de classification a été motivé par la possibilité d'une part de traiter séquentiellement les données, d'autre part de pouvoir mélanger des informations quantitatives et symboliques. Cependant, une analyse comparative approfondie de différentes méthodes de classification devrait être faite, de manière à mettre en place une approche utilisant plusieurs algorithmes en parallèle. Cette complémentarité pourrait permettre d'améliorer les résultats.

Au niveau de LAMDA il serait intéressant d'envisager la possibilité de pouvoir utiliser d'autres fonctions d'appartenance que celles actuellement disponibles. Et même, dans le cas où les descripteurs de l'ensemble de représentation n'ont pas le même type de distribution, il nous semble que la possibilité de choisir une fonction d'appartenance spécifique à chaque descripteur serait une extension pertinente à intégrer au niveau de SALSA.

Tout au long de notre travail, une des nos préoccupations principales, a été de présenter à l'expert des analyses et des résultats sous une forme facilement interprétable. Dans ce sens, une perspective de recherche au niveau de la génération d'un module d'interprétation des résultats de la classification devrait être initiée. Ces travaux pourraient dans un premier temps compléter les informations issues du profil de classes, en exploitant par exemple davantage le concept d'entropies intra-classe et inter-classe au niveau des descripteurs. Au niveau du regroupement de classes également, d'autres critères indépendants des données pourraient être mis en place.

Certaines de ces améliorations seront à court terme incluses dans le cadre du développement d'un outil professionnel de *Data Mining* reprenant certaines fonctionnalités de SALSA. Cet outil conçu indépendamment de tout contexte de supervision, est actuellement en cours de réalisation par le support R&D informatique du LAAS-CNRS.

D'autres perspectives peuvent être dégagées du modèle des états fonctionnels que nous avons proposé. En effet, la construction de l'automate pour la surveillance a montré la difficulté de valider les transitions par des concepts uniquement statistiques (comptages) alors que certains événements liés à des défaillances exceptionnelles, ne peuvent être observés qu'en très petit nombre, voire une seule fois. Dans ces

conditions, l’exploration de l’évolution des degrés d'adéquation globale (DAG) et des connaissances de l’expert sont à envisager.

Par ailleurs jusqu'à présent nous avons essentiellement utilisé l'automate pour la surveillance et plus précisément pour les fonctions de suivi et de détection de défaillances. Néanmoins, l'utilisation de l'automate à des fins de pronostic peut être envisagée dans la mesure où il serait possible à l'aide de l'automate de suivre des évolutions du processus conduisant de manière irrévocable vers une situation de défaillance. Ce type d'évolution, peut aussi être visualisé par les évolutions des adéquations globales des observations aux différentes classes.

Enfin, dans un cadre de supervision, l'automate peut apporter à l'opérateur des informations pour décider des actions de reconfiguration. L'automate en effet, fourni non seulement l'état de fonctionnement dans lequel se trouve le processus mais aussi des informations correspondantes aux descripteurs les plus impliqués dans les transitions observées.

Dans le futur, il apparaît donc intéressant de définir des processus de pronostic et de reconfiguration en s'appuyant sur le modèle des états fonctionnels que nous avons développé. Compte tenu du contexte dans lequel nous avons travaillé, la supervision de procédés chimiques et pétrochimiques, ces processus de pronostic et de reconfiguration devront être pensés dans un objectif *d'aide à la décision*.

Enfin, nous avons jusqu'à présent éludé les problèmes dus à l’éventuelle présence d’actions volontaires de commande liées à des événements de transition. Des travaux futurs sur cet aspect pourraient être amorcés et bénéficier des résultats sur la commandabilité des systèmes à événements discrets. Les problèmes relatifs à l'instrumentation pour la supervision et, plus particulièrement, au placement des capteurs sont également des perspectives dérivées de notre travail. Actuellement, des travaux de thèse orientés en ce sens [ORA05] sont menés.

BIBLIOGRAPHIE

[AGUA98] Aguado J.C., *A mixed qualiltative-quantitative self-learning classification technique applied to situation assessment in industrial process control.* Thèse de doctorat, Universitat Politècnica de Catalunya, 1998.

[AGUI80] Aguilar-Martin J., Balssa M., López de Mántaras R., "Estimation recursive d'une partition. Exemples d'apprentissage et auto-apprentissage dans R^n et I^n". Rapport technique LAAS-CNRS N° 880139, 1980.

[AGUI82] Aguilar-Martin J., López de Mántaras R., "The process of classification and learning the meaning of linguistic descriptors of concepts." *Approximate Reasoning in Decision Analysis*, North-Holland, pp.165-175, 1982.

[AGUI99] Aguilar-Martin J., "Knowledge-based supervision and diagnosis of complex process." *IEEE International Symposium on Intelligent Control, Intelligent Systems and Semiotics (ISI'99)*, Cambridge, USA, pp. 225-230, Septembre 1999.

[ASM] *ASM Consortium* disponible sur : http://www.iac.honeywell.com/pub/absitmang/.

[AYR95] Ayrolles L., Aguilar-Martin J., Guerrin F., "Intreprétation symbolique pour la supervision multi-résolution de porcessus complexes." *Supervision de processus à l'aides du Système Expert G2*, Hermes, pp. 73-90, 1995.

[BAB98] Babuska R., *Fuzzy Modeling for control,* Kluwer Academic Publishers, Massachusetts, USA, 1998.

[BAK92] Bakshi B., Stephanopoulos G., "Temporal representation of process trends for diagnosis and control." *IFAC symposium on online fault detection and supervision in the chemical process industry,* Newark, USA, 1992, pp.69-74.

[BEZ81] Bezdek J.C., "Pattern recognition with fuzzy objective function algorithms." *Plenum Publishing Corporation*, New York, 1981.

[BIS95] Bishop C., *Neural Networks for Pattern Recognition*, Oxford University Press, New York, USA, 1995.

[BRA96] Brachman R. J., Anand T., "The process of knowledge discovery in databases." *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, 1996

[CAS03] Casimir R., *Diagnostic des défauts des machines asynchrones par reconnaissance des formes.* Thèse de doctorat de l'École Central de Lyon, 2003.

[CASS97] Cassar J. P., Staroswiecki M., "A structural approach for the design of failure detection and identification systems". *Proceedings IFAC/IFIP/IMACS Conference on Control of Industrial Processe,* Belfort, France, pp. 329-334, 1997.

[CHA93] Chatain J. N., *Diagnostic par systèmes experts.* Hermes, Paris, 1993.

[CHE90] Cheung J.T., Stephanopoulos G., "Representation of process trends Part I. A formal representation framework." *Computers & Chemical Engineering,* Vol. 14, N° 4-5, pp.495-510, may 1990.

[CHEM] *Projet CHEM* disponible sur : http://www.chem-dss.org/

[CHO70] Chow C. K., "On optimum recognition error and reject trade-off," *IEEE Transactions on information theory*, vol. 16, pp. 41–46, 1970.

[COL01] Colomer J., Meléndez J., "A family of FIR differentiators based on a polynomial least squares estimation." *Proceedings of the european control conference*, pp.2802-2807, 2001.

[COL02] Colomer J., Meléndez J., Gamero F.I., "Pattern recognition based on episodes and DTW. Application to diagnosis of a level control system." *16th International Workshop on Qualitative Reasoning*, *QR 2002*, pp 37-43. Sitges, Barcelona, Espagne 10-12 juin 2002.

[COM00] Combacau M., Berruet P., Zamai E., Charbonnaud P., Khatab A., "Supervision and monitoring of production systems." 2nd management and control of production and logistics, MCPL, Grenoble, France, 2000.

[DAG01] Dague P., "Théorie logique du diagnostic à base de modèles". Chapitre 1 dans *Diagnostic, intelligence artificielle et reconnaissance des formes.* Hermes, Paris, 2001.

[DAS03] Dash S., Rengaswamy R., Venkatasubramanian V., "Fuzzy-logic based trend classification for fault diagnosis of chemical processes." *Computers and Chemical Engineering,* Vol. 27, pp. 347-362, 2003.

[DES87] Desroches P., *Syclare: Systeme de classification avec apprentissage et reconnaissance de formes. Manuel d'utilisation.* Rapport de recherche, Centre d'estudis avançats de Blanes, Espagne, Novembre 1987.

[DUB90] Dubuisson B., *Diagnostic et reconnaissance de formes. Traité des nouvelles technologies,* Série Diagnostic et Maintenance, Hermes, Paris, 1990.

[DUB93] Dubuisson B., Masson M., "A statistical decision rule with incomplete knowledge about classes," *Pattern Recognition*, vol. 26, pp. 155-165, 1993.

[DUD01a] Duda R., Hart P., Stork D., "Chapter 4: Nonparametric Techniques", dans *Pattern Classification*, Second Edition, Wiley-Interscience, 2001.

[DUD01b] Duda R., Hart P., Stork D., "Chapter 8: Non-metric Methods", dans *Pattern Classification*, Second Edition, Wiley-Interscience, 2001.

[ELT98] Eltoft T., deFigueiredo R., "A new neural network for cluster detection and labeling." *IEEE Transactions on neural networks,* Vol. 9, N° 1, pp. 1021-1035, 1998.

[ESC01] Escobet T., Travé-Massuyès L., Tornil S., Quevedo J., "Fault detection of a gas turbine fuel actuator based on qualitative causal models." *European Control Conference (ECC'01),* Porto, Portugal, pp. 2741-2746, Septembre 2001.

[FAY96] Fayyad U., Piatetsky-Shapiro G., Smyth P., "From Data Mining to Knowledge Discovery: An overview." *Advances in knowledge discovery and data mining,* AAAI Press/MIT Press, California, USA, 1996.

[FIS36] Fisher R. A., "The use of multiple measurements in taxonomic problems". *Annals of Eugenics*, Vol. 7, pp 179 – 188, 1936.

[FRA89] Frank P. M., Wünnenberg J., "Robust fault diagnosis using unknown input observer schemes", en: Patton R. J., Frank P.M., Clark R.N. (Eds.). *Fault diagnosis in dynamic systems – theory and applications.* Prentice Hall, London, 1989.

[GAL02] Galindo M., Aguilar-Martin J., "Interpretación secuencial de encuestas con aprendizaje LAMDA. Aplicación al diagnóstico en psicopatología." *Workshop IBERAMIA'02 - VIII Iberoamerican Conference on Artificial Intelligence*, Seville, Espagne, Novembre, 2002.

[GER98] Gertler J., *Fault Detection and Diagnosis in Engineering Systems.* Marcel Dekker, New York, 1998.

[GROWTH] *Competitive and Sustainable Growth Programme* disponible sur http://europa.eu.int/comm/research/growth/index.html

[GUS79] Gustafson D.E., Kessel W.C., "Fuzzy clustering with a fuzzy covariance matrix." *Proceedings of IEEE CDC*, San Diego, USA, pp. 761–766, 1979.

[HER94] Hérault J., Jutten C., "Perceptrons multicouches" dans *Réseau neuronaux et traitement du signal*, Traité des Nouvelles Technologies, série traitement du signal, Hermes, 1994.

[ISE93] Isermann R., "Fault diagnosis of machines via parameter estimation and knowledge processing", *Automatica,* Vol. 29, N° 4, pp. 815-835, 1993.

[ISE97] Isermann R., "Supervision, Fault Detection and Fault Diagnosis Methods – An introduction", *Control Eng. Practice,* Vol. 5, N° 5, pp. 639-652, 1997.

[JAI99] Jain A.K., Murty M.N., Flynn P.J., "Data clustering: A review." *ACM Computing Surveys,* Vol. 31, No. 3, September 1999.

[JAN91] Janusz M., Venkatasubramanian V., "Automatic generation of qualitative description of process trends for fault detection and diagnosis." *Engineering Applications of Artificial Intelligence,* Vol. 4, N° 5, pp.329-339, 1991.

[KAU90] Kaufman L., Rousseeuw P.J., *Finding Groups in Data: An Introduction to Cluster Analysis*, Wiley, New York, 1990.

[KEM03] Kempowsky T., Aguilar-Martin J., Subias A., Le Lann M.V., "Classification Tool based on Interactivity between Expertise and Self-Learning Techniques." *IFAC - Safeprocess 2003*, Washington D.C., USA, Juin 2003.

[KEM04] Kempowsky T., Subias A., Aguilar-Martin J., "Supervision of complex processes: strategy for fault detection and diagnosis." *MCPL 2004, Third Conference on Management and Control of Production and Logistics*, Santiago de Chile, Chile, 3-5 Novembre 2004.

[KOH95] Kohonen T., *Self-organizing maps.* Springer Series in Information Sciences, Vol. 30, 1995.

[KOT93] Kotch G., *Modular reasoning. A new approach towards intelligent control.* Thèse de doctorat, Swiss Federal Institute of Technology, Suisse, 1993.

[KUI86] Kuipers B., "Qualitative simulation", *Artificial Intelligence,* Vol. 29, N° 3, pp. 289-338, 1986.

[LUR01] Lurette C, Lalot S., Lecoeuche S. "Detection and Diagnostic of Drifts by Unsupervised and Auto-adaptative Neural Network", *Proceedings of the International Conference on Engineering Applications of Neural Networks, EANN 2001*, pp.239-246, Cagliari, Sardaigne, juillet, 2001.

[LUR03] Lurette C., Lecoeuche S., "Unsupervised and auto-adaptive neural architecture for on-line monitoring. Application to a hydraulic process." *Engineering applications of artificial intelligence,* Vol. 16, pp. 441-451, mai 2003.

[MAM77] Mamdani E., 3application of fuzzy logic to approximate reasoning using linguistic systems." *Fuzzy sets and systems*, vol. 26, pp. 1182-1191, 1977.

[MAR03] Marie-Joseph I., *Méthodologie de diagnostique appliquée à la maintenance préventive d'unités de production d'électricité en sites isolés.* Thèse de doctorat de l'Université des Antilles et de la Guyane, 2003.

[MCG95] MacGregor J. F., Kourti T., "Statistical process control of multivariate processes." *Control Engineering Practice,* Vol. 3, N° 3, pp. 403-414, 1995.

[MCQ67] MacQueen J., "Some methods for classification and analysis of multivariate observations." *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability,* Vol. 1, University of California, Berkeley, USA, pp. 281–297, 1967.

[MEL01] Meléndez J., Colomer J., "Episodes representation for supervision. Application to diagnosis of a level control system." *Workshop on Principles of Diagnosis DX'01,* Sansicario, Italy, 2001.

[MIC94] Michie D., Spiegelhalter D.J., Taylor C.C., *Machine Learning, Neural and Statistical Classification*, Ellis Horwood series in Artificial Intelligence, February, 1994

[MIT97] Mitchell T., "Decision Tree Learning", dans Mitchell T., *Machine Learning*, The McGraw-Hill Companies Inc., pp. 52-78, 1997.

[NAU95] Nauck D., Kruser R., "Nefclass – A neuro-fuzzy approach for the classification of data." *ACM Symposium on Applied Computing (SAC'95*), Nashville, USA, février 1995.

[NIM95] Nimmo I., "Adequately address abnormal situation operations", *Chemical Engineering Progress,* Vol. 91 No.9, pp. 36-45, 1995.

[NOM94] Nomikos P., MacGregor J., "Monitoring batch processes using multiway principal component analysis." *American Institute of Chemical Engineers Journal,* Vol. 40, N° 8, pp.1361-1375, 1994.

[ORA05] Orantes A., Le Lann M. V., Aguilar-Martin J., "Outil d'aide à la séléction des capteurs pour la détection de fautes et le diagnostic", *10ème congres de la societé française de genie des procédés, SFGP 2005,* Toulouse, septembre 2005.

[ORR96] Orr M., "Introduction to radial basis function networks." Rapport technique, Centre for Cognitive Science, University of Edinburgh, Écosse, 1996.

[PER98] Perez-Uribe A., "Artificial Neural Networks: Algorithms and Hardware Implementation." dans Mange D., Tomassini M., *Bio-Inspired Computing Machines. Towards novel computational architectures*, Presses Polytechniques et Universitaires Romandes, 1998.

[PIE89] Piera N., Aguilar-Martin J., "LAMDA : An incremental conceptual clustering method." Rapport technique LAAS-CNRS N° 89420, décembre 1989.

[PIE91] Piera N., Aguilar J., "Controlling Selectivity in Non-standard Pattern Recognition Algorithms." *IEEE Transactions on systems, man and cybernetics*, Vol. 21, N° 1, pp. 71-82, 1991.

[QUI86] Quinlan J.R., "Induction of Decision Trees", *Machine Learning*, Vol. 1, pp.81-106, 1986.

[REC66] Recht J. L., *Failure Mode and Effect.* National Safety Council, 1966.

[REI92] Reiter R., "A theory of diagnosis from first principles" en Hamscher W., Console L., de Kleer J.(Eds.), *Model-Based Diagnosis,* Morgan Kaufman, 1992.

[RIB02] Ribes J.C., Delaunay J., Delvaux J., Merle E., Moulliet M., "Diagnostic par reconnaissance de formes de l'état de fonctionnement de l'accélérateur AIRIX." *Journal Européen des Systèmes Automatisés, APII-JESA,* 2002.

[RUI01] Ruiz D., Nougés J.M., Puigjaner L., "Fault diagnosis support system for complex chemical plants." *Computers and Chemical Engineering,* Vol. 25, pp. 151-160, 2001.

[TAK85] Takagi T., Sugeno M., "Fuzzy identification of systems and its application to modeling and control." *IEEE Transactions on systems, man and cybernetics*, Vol. 15, N° 1, pp. 116-132, 1985.

[TOG00] Toguyéni A. K. A., Craye R., "Alarm processing and faults recovery based on functional modeling." *Proceedins of World Automation Congress*, Maui, Hawaii, June, 2000.

[TRA97a] Travé-Massuyès L., Dague P., Guerrin F. (dir.), *Le raisonnement qualitatif pour les sciences de l'ingenieur*, Hermes, Paris, 1997.

[TRA97b] Travé-Massuyès L., Milne R., "TIGERTM : gas turbines condition monitoring using qualitative model based diagnosis." *IEEE Expert. Intelligent systems & their applications*, Vol. 12, N° 3, pp. 22-31, Mai-Juin 1997.

[UCI98] Blake, C.L., Merz, C.J., *UCI Repository of machine learning databases* [http://www.ics.uci.edu/~mlearn/MLRepository.html]. University of California, Department of Information and Computer Science, Irvine, USA, 1998.

[VAN03] Van de Steene L., Girard P., "Le point sur la gazéification de la biomasse." *Bois Énergie*, N°1, mars 2003.

[VED97] Vedam H., Venkatasubramanian V., "A wavelet theory-based adaptive trend analysis system for process monitoring and diagnosis." *Proceedings of the American Control Conferences*, Piscataway, USA, pp.309-313, 1997.

[VEN03a] Venkatasubramanian V., Rengaswamy R., Yin K., Kavuri S., "A review of process fault detection and diagnosis. Part I: Quantitative model-based methods", *Computers & Chemical Engineering*, Vol. 27, N° 3, pp. 293-311, 2003.

[VEN03b] Venkatasubramanian V., Rengaswamy R., Kavuri S., Yin K., "A review of process fault detection and diagnosis. Part III: Process history based methods." *Computers & Chemical Engineering*, Vol. 27, N° 3, pp. 327-346, 2003.

[VEN90] Venkatasubramanian V., Vaidyanathan R., Yamamoto Y., "Process fault detection and diagnosis using neural networks I: steady state processes." *Computers and Chemical Engineering,* Vol. 14, N° 7, pp. 699-712, 1990.

[VIL88] Villemeur A., *Sûreté de fonctionnement des systèmes industriels. Fiabilité Facteurs humains Informatisation.* Eyrolles, 1988

[WAI00] Waissman-Vilanova J., *Construction d'un modèle comportemental pour la supervision des procédés : application à une station de traitement des eaux.* Thèse de doctorat, Institut National Polytechnique de Toulouse, 2000.

[WAI98] Waissman J., Aguilar J., Dahhou B., Roux G., "Généralisation du degré d'adéquation marginale de la méthode de classification LAMDA", *6èmes rencontres de la Société Francophone de Classification*, Montpellier France, 1998.

[WAI99] Waissman J., Sarrate R., Dahhou B., Aguilar-Martin J., "Building an Automaton for Condition Monitoring in a Biotechnological Process." *5th European Control Conference*, Karlsruhe, Germany, 1999.

[YAG94] Yager R. R., Filev D. P., "Chapter 5: Fuzzy system models" dans *Essentials of fuzzy modeling and control.* John Wiley & sons, Inc., 1994.

[ZAD73] Zadeh L., "Outline of a new approach to the analysis of complex systems and decision processes." *IEEE Transactions on systems, man and cybernetics,* SMC 3, pp. 24-44, 1973

[ZAD78] Zadeh L. A., "Fuzzy sets as a basis for a theory of possibility." *Fuzzy Sets and Systems*, Vol. 1, pp. 3-28, 1978.

[ZAM98] Zamaï E., Combacau M., Chaillet-Subias A., "Models and strategies for monitoring of flexible manufacturing systems." *9th symposium on information control in manufacturing,* Nancy-Metz, France, juin 1998.

[ZWI95] Zwingelstein G., *Diagnostic des défaillances - Théorie et pratique pour les systèmes industriels.* Traité des Nouvelles Technologies, série Diagnostic et Maintenance. Hermès, Paris, 1995.

ANNEXE A.

PROJET EUROPEEN CHEM

Le projet CHEM a été financé par la *Communauté Européenne* dans le cadre du programme "*Croissance Compétitive et Durable*" du 5° Programme cadre pour des actions communautaires de recherche, de développement technologique et de démonstration (*RTD - Research and Technological Development*) (1998-2002) [GROWTH] sur contrat G1RD-CT-2001-00466.

A.1 La problématique – objectifs

L'objectif principal du projet *"**CHEM** : Advance Decision Support System for Chemicla/Petrochemical Manufacturing Processes"*, était de développer et mettre en place un ***Système d'Aide à la Décision*** (SAD) pour la surveillance, l'analyse et l'interprétation des données, la détection et le diagnostic des événements, et le support aux opérateurs dans la prise de décisions dans l'industrie chimique et pétrochimique.

Ce système est construit à partir de l'intégration synergique des outils logiciels (*toolbox*) basées sur des nouvelles technologies que permettent la détection et le diagnostic robuste, en temps réel des problèmes du processus. L'ensemble permettra d'améliorer la sûreté, la qualité du produit et la fiabilité d'opération ainsi que de réduire les pertes économiques dues à des situations de défaillances principalement, dans le cadre de procédés chimiques et pétrochimiques. Ce système a comme but principal d'aider des opérateurs dans l'évaluation et l'identification des situations du processus et de répondre aux événements anormaux, permettant, de ce fait, aux installations de maintenir à coût réduit l'intégrité opérationnelle et d'améliorer la qualité de ce qu'elles produisent.

L'idée principale était donc de développer et construire un système complet pour augmenter les possibilités d'une gestion sans risque et plus efficace des usines chimiques et pétrochimiques. Un tel outil a une portée extrêmement large, englobant la gestion d'information, la détection et le diagnostic des défaillances, la gestion des alarmes et l'optimisation.

A.2 Le contexte industriel

Les principales motivations du projet CHEM concernaient :

- Les **performances économiques,** qui ne peuvent pas tolérer de longs arrêts ou la présence de situations anormales : le gain obtenu par la *commande avancée de procédé* (CAP) qui essaye de garder le processus le plus proche possible d'un point optimal, peut être perdu lors de situations anormales. Cette perte économique est estimée en milliards d'euros par an, même en ne prenant en compte que les dommages qui peuvent être provoqués par de très grands accidents.

- Les soucis **environnementaux** et de **sûreté** qui impliquent de plus en plus de contraintes. Les industries doivent assurer une fabrication propre de leurs produits avec un cycle de vie acceptable en termes de protection de l'environnement et de

gestion des ressources naturelles disponibles. Il leur faut minimiser la pollution ainsi que les accidents et optimiser les procédures opératives pour éliminer le gaspillage des matériaux et d'énergie, tout en respectant les normes de qualité et environnementales de plus en plus exigeantes.

- La **flexibilité** est absolument nécessaire : pour pouvoir rester compétitives, les compagnies doivent réagir aux changements du marché et répondre aux demandes des nouveaux produits.

- La **complexité** des systèmes de commande rend de plus en plus difficile la prise de décisions. L'avancée technologique sur les capteurs, les systèmes de commande distribuée et l'informatique ont augmenté la quantité de données rassemblées sur les unités industrielles et présentées aux opérateurs qui doivent prendre les décisions importantes et stratégiques. Cette information doit être employée pour améliorer l'efficacité et la productivité, pour éviter des arrêts non programmés ainsi que des situations anormales. En donnant des outils appropriés aux opérateurs, le projet CHEM améliorera leurs conditions de travail. Les outils basés sur l'intelligence artificielle fournissent des explications dont la plupart sont facilement interprétables, ce qui permet d'augmenter le niveau de connaissance sur le processus.

Ces différentes préoccupations mènent à la conclusion qu'il y a un besoin industriel réel pour des méthodes visant à améliorer la gestion des processus : des fonctions intelligentes pour les systèmes avancés d'aide à la décision sont nécessaires.

Beaucoup de travail a été conduit dans des domaines semblables : analyse de tendances, diagnostic de défaillances et systèmes d'aide à la décision, mais les méthodes développées fonctionnent toujours séparément et il reste difficile, pour les concepteurs des systèmes industriels de supervision, de construire des applications adaptées aux nouvelles usines.
Par conséquent, il est nécessaire de rassembler les efforts des spécialistes avec une gamme étendue d'expertise pour pouvoir concevoir un système complet de supervision.

Il existe parmi les outils commerciaux de développements génériques des tentatives de mise en place de systèmes d'aide à la décision avancés. Nous pouvons citer les développements réalisés en G2 pour Lafarge, Amoco et Mobil. Néanmoins, il s'agit d'applications pour la surveillance des procédés très particulières et il est difficile de généraliser de telles expériences à d'autres applications.

Par ailleurs, des initiatives similaires à celle du projet CHEM existent aux États-Unis, telles que, le consortium *"Abnormal Situation Management (ASM)"* mené par Honeywell [ASM], et le *"Programme de Contrôle Intelligent"* du NIST. Ces initiatives ont toutefois des objectifs un peu différents. L'ASM met l'accent sur l'interaction homme-machine et la personnalisation des systèmes construits autour des architectures Honeywell, pour permettre aux opérateurs de prévenir des situations anormales. Le but du programme du NIST est de développer et valider des solutions intégrées pour la commande de processus de fabrication.

A.3 Organisation et partenaires

Les partenaires du projet proviennent de la communauté européenne, du Japon et de la Corée. Les participants sont des établissements académiques et des industriels qui fournissent des logiciels ou disposent de sites industriels pour valider les *"toolboxes"*.

Le consortium européen a été établi de sorte que les partenaires puissent fournir des techniques complémentaires et différentes applications représentatives. Ont participé à ce projet, cinq institutions académiques de huit pays : l'Université Polytechnique de Catalunya (UPC), l'Institut Technologique de Lund, l'Université de Girona et le LEA-SICA, l'Université de Lille (USTL-LAIL) et l'Université de Technologie de Varsovie (WUT) et 9 partenaires industriels : l'Institut Français de Pétrole (IFP), Gerth (spécialiste en gestion financière), Thalès, Computas, VTT (spécialiste en gestion de risque), UPM-Kymmene, KCL, Corus-British Steel Limited et Gensym.

Ce projet international aborde différents aspects qui sont intégrés dans une architecture modulaire dont différents modules ("toolboxes") peuvent collaborer. Le consortium européen s'est centré sur les aspects suivants :

- L'interprétation des informations recueillies par des capteurs : déterminer la fiabilité des données, suivre et analyser les tendances du processus, faire une abstraction des signaux.
- La détection des défaillances et la gestion d'alarmes afin d'éviter des situations anormales : déterminer les causes des événements (diagnostic) et déterminer leurs conséquences.
- La planification réactive et le développement de structures de décision afin de déterminer le nouvel état du procédé à atteindre, en prenant en compte les événements détectés et les capacités restantes du procédé.

La structure du projet international comprenait 12 "*Work Packages (WP)*" à savoir :

WP1 : Conception générale et Méthodologie d'Intégration.
WP2 : Prétraitement et conciliation des données
WP3 : Analyse des tendances, qualité, identification et évaluation des situations du processus.
WP4 : Évaluation de performance de systèmes de commande
WP5 : Diagnostic de défaillances et gestion d'alarmes.
WP6 : Optimisation de processus
WP7 : Structures d'aide à la décision
WP8 : Réglage du système de commande
WP9 : Planification et ordonnancement réactifs
WP10 : Intégration des "Toolboxes"
WP11 : Tests et validation industrielle
WP12 : Gestion projet

Les partenaires européens ont participé à 8 d'entre eux (WP1, WP3, WP5, WP7, WP9, WP10, WP11 et WP12). La figure A-1 décrit l'architecture et la plateforme d'intégration pour la mise en place du système. WP1 définit la façon dont les *toolboxes (TB)* seront conçues et intégrées et fournit une méthodologie de développement. WP3…WP9 groupent les différents *TBs* qui fournissent les fonctions décrites précédemment. WP10 vérifie et assure l'intégration entre les *TBs*. Le WP11 est chargé de conduire les tests relevants et la validation des TBs dans des unités industrielles. Finalement, la gestion administrative, la coordination et l'exploitation du projet sont conduites par le WP12.

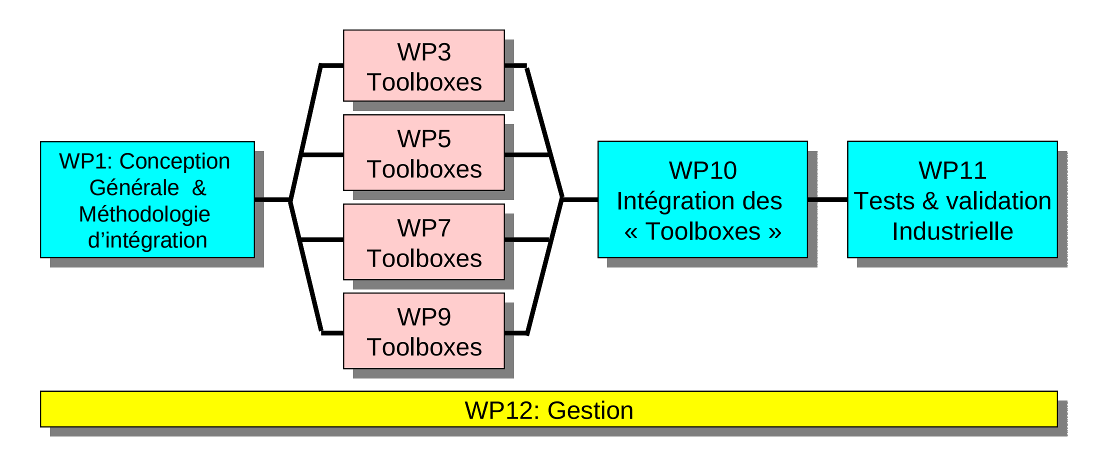


Figure A-1. Plate-forme d'intégration et d'architecture du projet CHEM

A.4 Notre participation : contexte contractuel

Notre participation dans le projet CHEM se situe au sein du WP3 – Analyse des tendances, qualité et identification des situations du procédé. L'objectif de ce WP est de mener l'identification de l'état courant du procédé, par le biais de l'analyse des tendances des variables, en prenant avantage des méthodes quantitatives, qualitatives et semi-qualitatives (intervalles) basées sur des modèles imprécis ou simplement à partir des données historiques. L'analyse des tendances ainsi que l'identification des situations est réalisée à partir de l'extraction des informations quantitatives et qualitatives des mesures issues du procédé et de leur représentation facilement interprétable par l'opérateur humain. Le but des outils de ce WP est de déterminer l'état normal ou anormal d'une variable et d'identifier la situation dans laquelle se trouve le procédé. L'idée est d'intégrer les résultats des TBs du WP3 avec les TBs des WP5 (diagnostic et gestion d'alarmes) et WP7 (structures pour la décision).

Les TBs comprises dans le WP3 sont :

TB3.1 : Représentation qualitative des tendances du processus (Episodes)
TB3.2 : Génération de séquences de bandes temporelles
TB3.3 : Décomposition des tendances du processus

TB3.4 : Génération d'alarmes
TB3.5 : Modélisation du procédé à partir de logique floue et réseaux de neurones
TB3.6 : Identification qualitative des situations du procédé en ligne
TB3.7 : Gestion de qualité et performance basée sur la commande statistique multi-variables (Analyse des Composantes Principales).

Nous avons eu comme tâche le développement de la ***TB3.6***. L'activité principale de cette TB était de transformer les mesures quantitatives et les informations qualitatives provenant d'autres TBs, en une représentation qualitative des situations du procédé définies par l'utilisateur (opérateur expert). Le but était donc d'intégrer les résultats des TB3.1 (épisodes), TB3.3 (décomposition des tendances) et TB3.7 (composantes principales), afin d'obtenir des situations ou des états qualitatifs ou semi-qualitatifs, pour représenter le comportement du procédé défini par un ensemble de variables.

Cet outil a été nommé SALSA (*Situation Assessment using LAMDA claSsification algorithm*). SALSA réalise deux activités principales. La première est exécutée hors-ligne et comprend la construction d'un système de classification avec l'apprentissage et la génération des classes à partir des données historiques du processus, et l'affectation de ces classes aux différents états fonctionnels identifiés. Cette partie repose sur un dialogue permanent avec l'expert. La deuxième activité est la reconnaissance en-ligne de l'état dans lequel le processus se trouve, ceci en utilisant l'information issue des capteurs et actionneurs du procédé ou d'autres *toolbox* ainsi que le système de classification conçu lors de la phase d'apprentissage.

A.5 Utilisateurs finaux : les unités industrielles

Les différents procédés qui ont étés fournis par les partenaires industriels correspondent à un ensemble assez varié de processus qui ont permis, non seulement la validation des performances des TBs, mais aussi plusieurs configurations d'intégration des différents outils, afin de trouver la plus adaptée aux besoins et contraintes de chaque procédé. Ces unités industrielles variées sont : une unité de cracking catalytique, un processus de fabrication de papier, une unité pilote de gazéification de biomasse, un générateur de vapeur, un haut fourneau et une usine d'épuration de gaz (Coke oven gaz cleaning plant).

Dans le cadre du projet, nous avons été amenées à tester notre travail sur trois de ces unités (le gazéifieur de biomasse, l'usine d'épuration de gaz et le générateur de vapeur). Les résultats obtenus ont été présentés dans le chapitre 5 de cette thèse.

ANNEXE B.

MANUEL D'UTILISATION DE SALSA



User's Manual

SALSA – User's Manual February 2004 v1.2

Contents

Figures Table

1. ABOUT THIS USER MANUAL

This manual is a reference for the qualitative situation assessment of a given industrial process. It provides the guidelines for the construction of classification systems for the offline characterisation and online identification of the functional states of a process by means of a learning classification technique using qualitative and/or quantitative information of the process variables and the expert's knowledge.
The manual is divided into two main chapters as follows:

Chapter 2: It covers some technical information, the installation of the application and the system requirements.

Chapter 3: This section provides sufficient information to use the application. It shows an overview of the basic features of SALSA as well as the steps that must be followed.

SALSA application as well as this user manual were developed by the DISCO group from the LAAS-CNRS.
For further information and details please contact: Joseph AGUILAR-MARTIN (aguilar@laas.fr), Tatiana KEMPOWSKY (kempowsky@laas.fr).

This is a January 2004 version 1.1 of SALSA User's Manual. Updates will be constantly made.

2. SOFTWARE INSTALLATION

2.1 Introduction

The main function of SALSA is to identify in a qualitative way the current functional state of a supervised process using the online symbolic and/or numeric information from the process variables. To do this SALSA has two main stages. The first, offline stage corresponds to a *learning stage* which, from a given data set, generates classes and assigns significant functional states to these classes by a constant dialogue with the expert. The learning stage includes:

- Class Generation: a clustering classifier with the capacity to perform a supervised or unsupervised learning procedure. Interaction with the expert is necessary in order to tune up the classifier parameters.
- Mapping Classes to States: a dialog tool with the expert, for the assignment of classes, or groups of classes, to meaningful functional states.

The online stage is the *recognition stage*, which will determine de functional state of the process from online data of plant sensors and actuators or other information from the process variables.
During plant operation SALSA generates the current functional state of the supervised process by using real-time (online) data.

2.2 Technical Data

- This version of SALSA was programmed in plain ANSI-C, although the underlying philosophy is object-oriented.
- SALSA is a standalone tool, it can operate on its own without need of another software tool.
- The selected platform for the application's development was LabWindows/CVI 6:
  - Graphic User Interface
  - Source Code programming
- The software may be installed in Windows 98/NT/2000/XP

2.3 SALSA Setup

1. From the installation Folder <u>InstallSALSA04vbat</u> double click at the '**setup.exe**' file and let you guide by the installation wizard.
2. Once SALSA application has been installed verify that the directory SALSA04v1 has been created.
3. Verify that the sub-folder **\DataTr** has been created in the SALSA04v1 directory. This folder is required for the online data transfer between SALSA and another application running online.

3. GETTING STARTED

3.1 SALSA MAIN SCREEN

When SALSA is launched, Figure **1** shows its principal window. There are three main options to choose:

- **<u>OFFLINE</u>**: Learning Stage. This concerns the design and construction of a classification system, from a set of experimental data. It consists of two main stages: a learning stage (unsupervised or supervised) and the assignment of the resulting classes into significant functional states. Active participation of the process expert is required.

- **<u>ONLINE</u>**: Recognition Stage. Using a previously generated classification system, the current functional state of the process will be identified.

- **<u>QUIT</u>**: This option exits the toolbox. To exit the application the user must first close (quit) the option he is working on (one of the above) an exit by clicking the QUIT button in the principal MENU window.

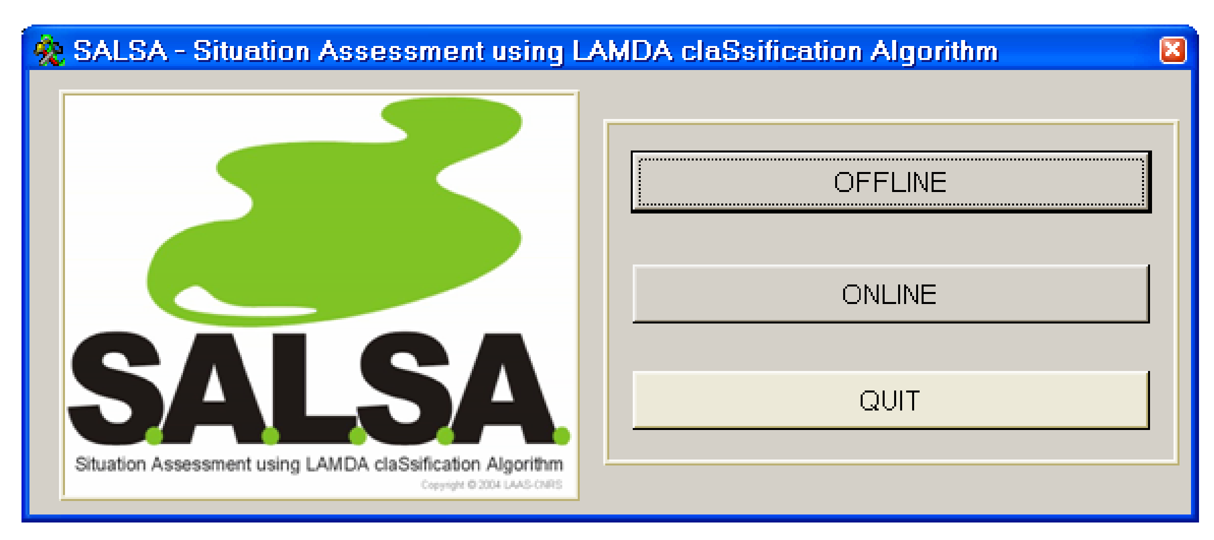


Figure 1 - SALSA - Main Screen: Options

3.2 SALSA – OFFLINE

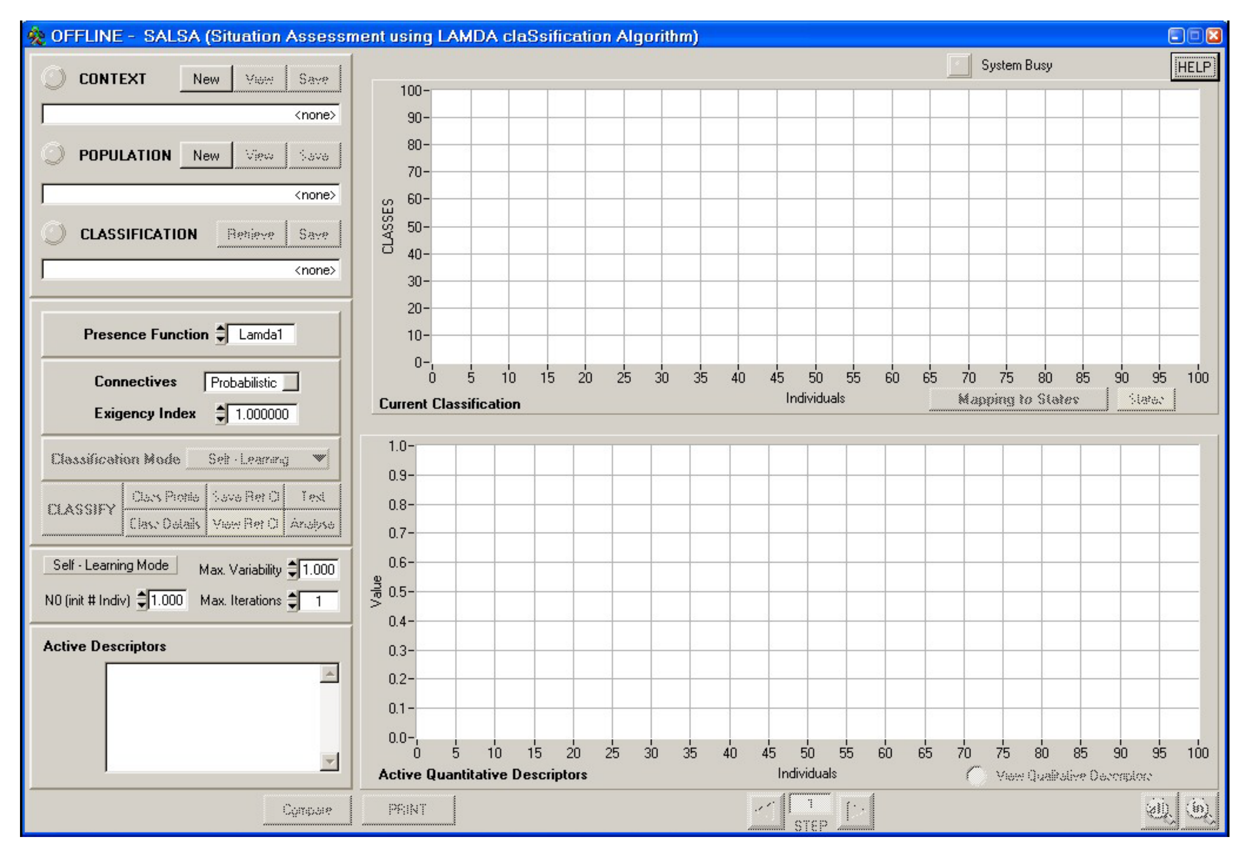


Figure 2 – SALSA - OFFLINE Main Screen

Figure **2** shows the main screen for SALSA offline (learning) stage. The following sections give some guidelines of the features and use of this stage.

3.2.1 Functions of the Offline stage

3.2.1.1 Loading of the training data file in the specified format

- **Format Specification**: The format for the input (training) files (the context data base and the population to be classified) is a text file with a .DAT extension. Lines can be blank ones, comments (MatLab style, always preceded by "%") or observations. These observations or **individuals** present values for the *context descriptors* in columns separated by **'blank spaces'** and may have an additional field if we are interested in supervised learning (i.e. the additional field correspond to the pre-defined class).

  - Type of Data: a descriptor is assumed to be ***quantitative*** if it starts with a number, a point or a minus sign, and ***qualitative*** otherwise.

  - The name or tag: for each descriptor must be a line with the "&" sign at the beginning of the line, and they should be also separated by a 'blank space'.

  - Pre-defined classes: the additional field for supervised learning will indicate the class where the individual is to be assigned. The algorithm expects these numbers to appear in a semi-ordered way, that is, the class "1" has to appear at least once before the class "2", there has to be at least one "2" before any "3" and so on. When these numbers are greater than 0, they create a class. The zero (0) means that the individual is part of a population that later will undergo pattern recognition. Figure 3 shows an example of the training file format for the case of supervised learning.

```
% LEVEL.DAT
%
% This individual file collects data from a real pilot plant system
% which consists of two tanks and a control system to manage the
% level of the second one.  This data base has been generated from
% three possible situations: normal operation (class one), a partial
% obstruction between tanks (class two), and the emergency situation,
% total obstruction at the exit of tank two (class three).
%
% Used descriptors are: level in tank one, level in tank two (both from 0 to
% 10), the desired output or control input (only High or Low), the control
% signal (also from 0 to 10), and the derivatives of first, second level and
% control signal (coded: Negative Large, Negative Small, Zero,
% Positive Small and Positive Large).
%
% We do have a last field of imposed class.

&LEVEL_T1 LEVEL_T2 OUTPUT CTRL_SIG DERIV_T1 DERIV_T2 DERIV_CONTROL

9.107939 3.021866 A 5.622508 D D C 1
9.312390 2.988349 A 5.940937 B C D 1
7.678884 1.027721 C 4.984652 A B D 1
9.994041 2.507010 A 4.953032 D A E 1
5.344999 2.015358 C 3.481344 E E A 1
9.031239 3.020050 A 5.714987 D D B 0
8.579471 1.061223 C 4.925020 D B D 2
8.196658 0.961353 C 6.249512 E D B 2
8.619834 0.909940 C 6.692107 A D B 0
7.391809 0.987452 C 5.994978 E D B 2
8.945312 3.056641 A 5.458984 B D B 1
6.353343 5.844610 A -1.251475 E A C 1
5.928058 6.129644 A -1.275735 C C C 3
5.927319 6.129463 A -1.276346 D C C 3
5.955299 6.138647 A -1.276663 C C C 0
5.958092 6.142730 A -1.283172 D C C 3
7.255859 0.976562 C 5.537109 D B C 1
```

Figure 3 – Example of the .DAT file

- **Loading a Training File:** To load a file the user must select either the 'New Context' or the 'New Population' buttons (Figure 4). Although it is possible to start with any of the two options it is recommended to first load the context data file and then the population file.



Figure 4 – New Context & New Population buttons

- Loading a new **context:** The context file normally contains the number of descriptors and its type (numeric or symbolic). It may be induced from a population (or plain text) file with a .DAT extension or retrieved from a binary file previously saved with a .CON extension (Figure 5). If the context has been 'induced' from a population the application will demand the user if he wants also to load the corresponding population to be used for classification.

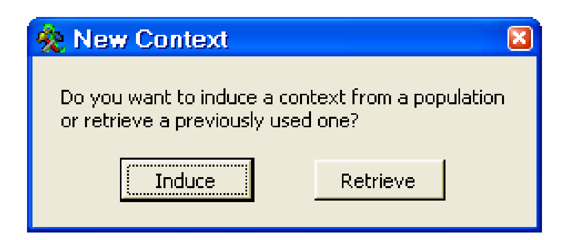


Figure 5 - Loading a New Context: Induce or Retrieve

- Loading a new **population**: This file contains the individuals that will be used for training. It must have the same number and type of descriptors as in the context and it may contain an additional field corresponding to the pre-defined classes (to be used for the supervised learning case). A dialog window will appear asking the user if he wants to load the corresponding context (Figure 6). If changes have previously been made (e.g.: a descriptor has been disabled) to the context, they will be lost if the user accepts loading the new context also.

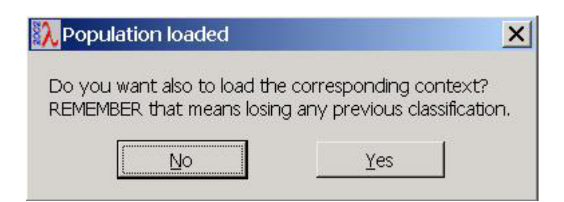


Figure 6 – Loading a 'New' population

- Because of the presence of commented lines (with a '%' symbol) and a possible supervised learning field, there exist a couple of dialog windows that ask the user if the population file is to be looked at or edited with a Windows text editor, and if the last field is to be treated like an imposed classification value. See Figure 7.



Figure 7 - View/Edit File & Imposed Classification Windows

3.2.1.2 Selection of the Classification Algorithm - Quantitative descriptors

To make the different calculations required in the learning and classification procedures it is necessary to choose the type of algorithm to be used for the quantitative descriptors (Figure 8):

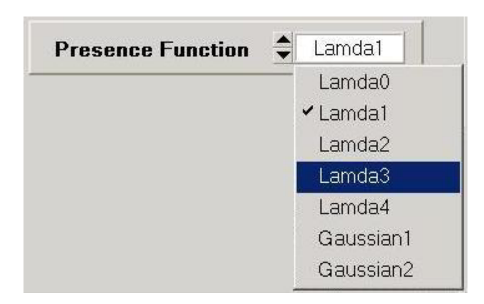


Figure 8 - Presence Function selection

- Lamda1 : $\rho^{x}(1-\rho)^{1-x}$
- Lamda2 : $K\rho^{x}(1-\rho)^{1-x}$ $\qquad K=\dfrac{\log\left(\frac{\rho}{1-\rho}\right)}{2\rho-1}$
- Lamda3 : $\rho^{1-|x-c|}(1-\rho)^{|x-c|}$
- Lamda4 : $K\rho^{1-|x-c|}(1-\rho)^{|x-c|}$ $\qquad K=\dfrac{\log\left(\dfrac{\rho}{\rho-1}\right)}{2\rho-\rho^{c}(1-\rho)^{1-c}-\rho^{1-c}(1-\rho)^{c}}$
- Gauss1 : $e^{-\frac{(x-\mu)^2}{2\sigma^2}}$
- Gauss2 : $Ke^{-\frac{(x-\mu)^2}{2\sigma^2}}$ $\qquad K=\dfrac{\sigma_0}{\sigma}$

3.2.1.3 Selection of the connectives and the exigency level

- Connectives: The user may choose between two different families of logic operators (T-norms & dual T-conorms functions) in order to aggregate all the Marginal Adequacy Degrees of an individual to a class (Figure 9). These operators are: Product-Probabilistic Sum (T-norm & T-conorm) and the Minimum-Maximum (T-norm & T-conorm).

- Exigency Level: The user may also use mixed connectives of the same family by choosing an exigency level (α) between 0 and 1 (Figure 9). In recognition we call α the *exigency level* and in self-learning it is called *selectivity level*. By changing the value of α different partitions based on the same data used may be obtained. Thus, as the value of α increases more classes will be created or in the case of recognition a greater adequacy is required of a measurement to thee assigned to a pre-established class.



Figure 9 – Connectives & Exigency selection

3.2.1.4 Selection of the classification mode

When a file has been loaded for the first time the classification mode will be selected according to the characteristics of the file.



Figure 10 – Classification Mode selection

- **Self – Learning**: Unsupervised learning (No pre-defined classes). There is no preliminary knowledge. SALSA creates a group of classes using as base the values of the descriptors from the individuals of the data file that has been loaded. There are two selectors that may be changed by the user only when unsupervised learning has been chosen. These are the maximum desired variation percentage and the maximum allowed iterations number (see Figure 11).



Figure 11 – Max. Variability & Iterations selectors

They are necessary because in unsupervised learning the class parameters vary, so to obtain some stability and to overcome in a certain way the effect of the observations ordering, the same population is classified several times until a maximum percentage of individuals changes from one iteration to another. However the time to reach stability could be very long, so a maximum number of iterations has also been introduced. This means that the population will be presented to the existing classes once and again until either no more than the specified percentage of individuals varies its assignment from the previous iteration or the cycles limit has been reached.

- **Supervised Learning**: The loaded population has a class imposition field. This option allows performing a different number of choices, like learning from an initial set of classes, which can be modified by adding new classes or by updating their parameters or both (Figure 12).



Figure 12 – Supervised Learning Options window

- **Pattern Recognition**: This option has two alternatives, either the user allows unclassified individuals, meaning that an individual has not been recognized in any class (its adequacy degree is lower than the minimum threshold) and has been placed in the NIC class, or force every individual to be assign to a class, in this last case the Non-Informative Class is not taken into account for recognition (Figure 13).

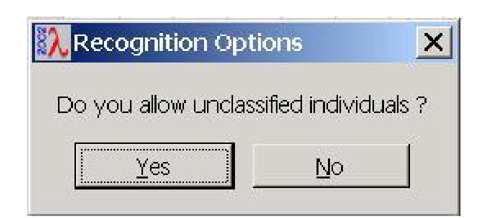


Figure 13 – Recognition Options window

3.2.1.5 Data Processing, Treatment and Visualisation

- Once the context and the population have been loaded, SALSA identifies the number of descriptors and its nature, and it internally rescales quantitative values between 0 and 1 ("Normalisation" process).

- To make possible a direct confrontation between individuals and classes, it is necessary that the descriptors used for observations are the same (same order) than the concepts used for classes (context). However, whenever the number of descriptors, or its nature, or the actual values exhibited in the population do not fit the ones stored in the context, the problem is reported in an error file, named **"...\error.out**". The error is explained in this file with an indication of its nature type and location (number of individual and number of descriptor if necessary). A message window is displayed to notify the user. If the population can still be loaded, for instance, if a value is lower than the minimum value SALSA will take into account a 0 or if a value is greater than the maximum value SALSA will replace it with a 1, the process will continue.

- The user may view each context descriptor separately and in a detailed way, as well as the values of each individual (Figure 14). It is possible for the user to change the descriptors he wants to use for learning and classification; he can *enable* and *disable* a descriptor for classification.

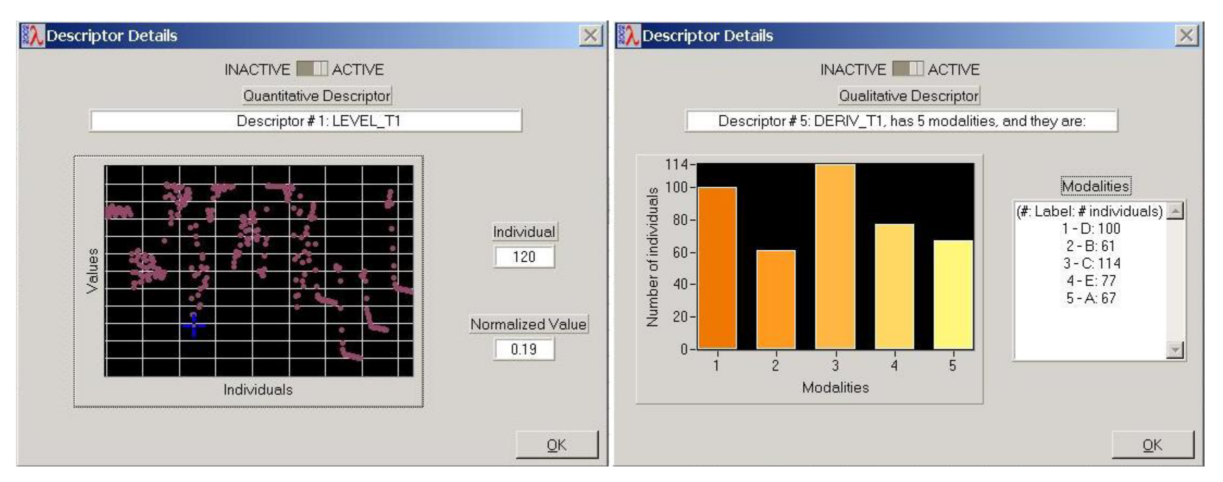


Figure 14 - Quantitative & Qualitative Descriptor windows

For qualitative descriptors, a histogram of the number of individuals for each modality is shown in Figure 14 as well as a small window that gives the name of the modality and the exact number of individuals. For quantitative descriptors a graph representing each individual with its normalized value is shown.

There is also a text window with all the information about the loaded context (see Figure 15).

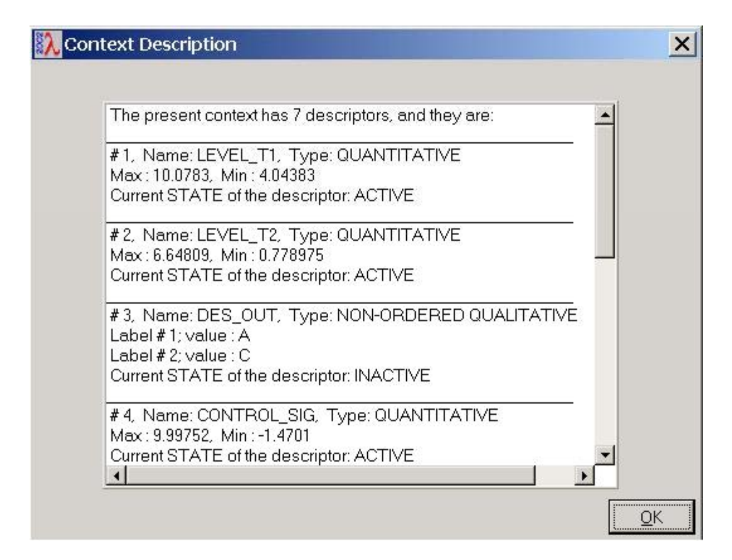


Figure 15 – Context Description window

- By clicking the **'View' Population** button all active descriptors are graphically displayed. There is a graph for quantitative descriptors where for each individual its normalized value is shown, and at the left part of the graph the names of the active quantitative descriptors are displayed. For qualitative descriptors, modalities corresponding to each individual are shown in another graph. The user may change from one graph to another using the corresponding option: 'View Qualitative Descriptors" or "View Quantitative Descriptors", which is located at the right bottom corner of each graph. (See Figure 16 & Figure 17).

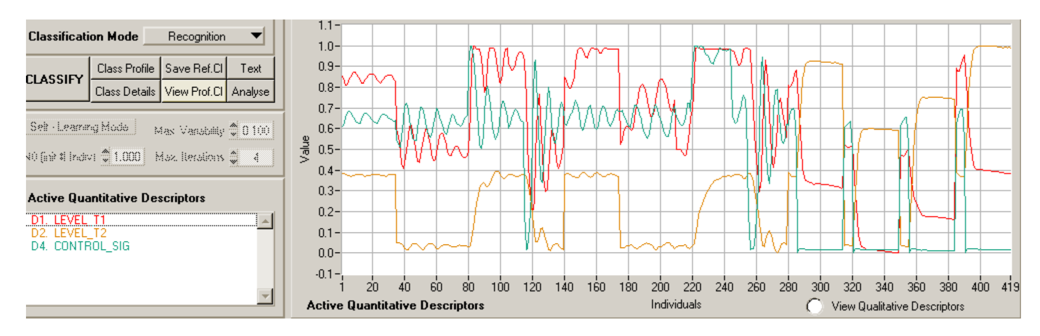


Figure 16 – Active quantitative descriptors graph

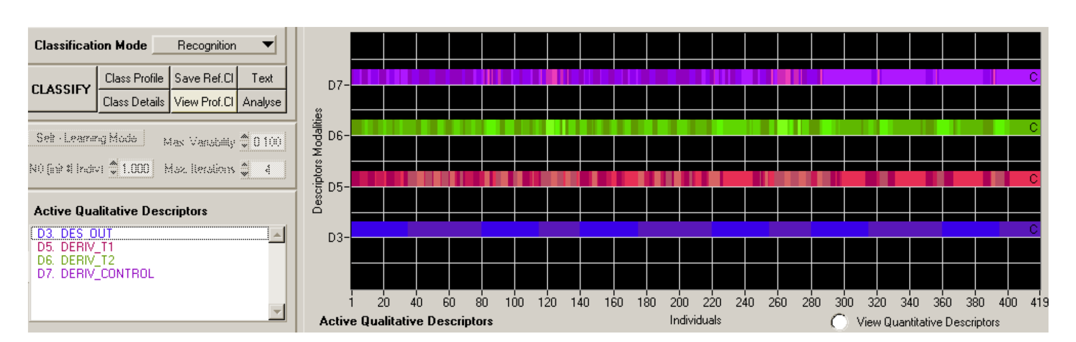


Figure 17 – Active qualitative descriptors graph

For qualitative descriptors we represent each modality with a similar but different colour.

3.2.1.6 Classification Results and Visualisation

- After successfully loading context and population (green LEDs for Context and Population and name of the loaded file), it is possible to put the algorithm to work by pressing the **Classify** button. See Figure 18

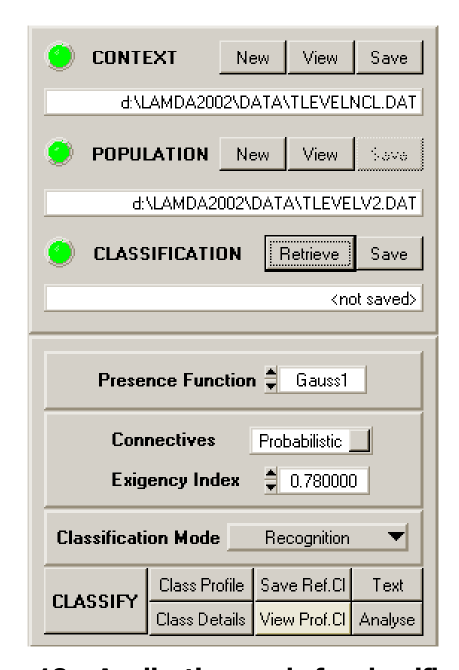


Figure 18 – Application ready for classification

- Once a classification has been made, SALSA graphically displays the class assigned to each individual (**Figure** 19), and by clicking on the **Class Profile** button (Figure 20)the profile of each class will be displayed.

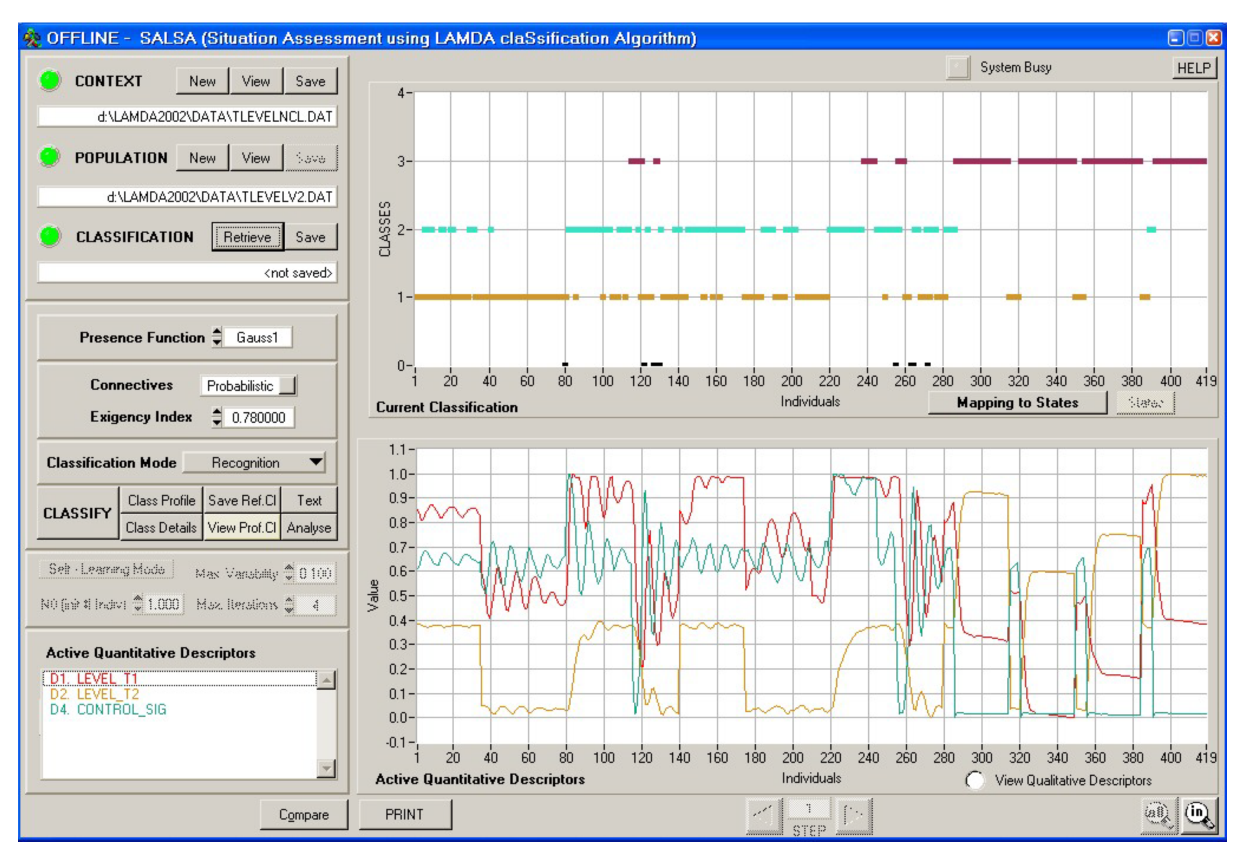


Figure 19 – Classification Results using Supervised Learning



Figure 20 – Classification visualizing buttons

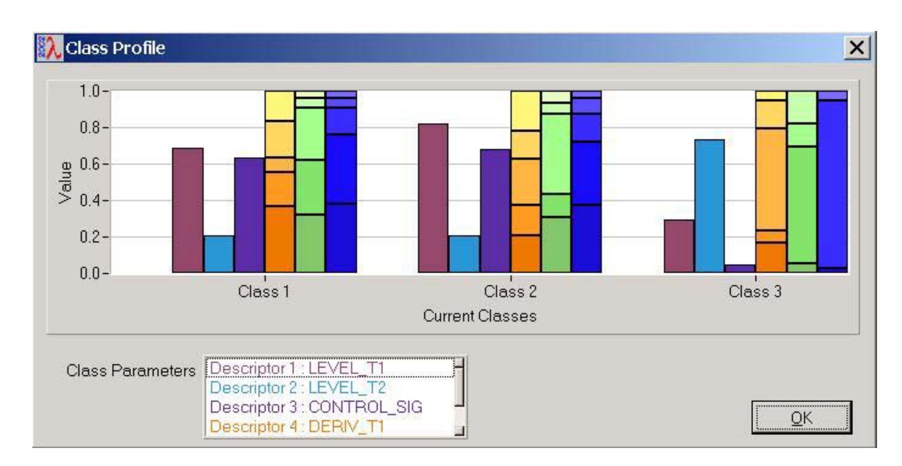


Figure 21 – Profile for each Class

Figure 21 shows the profile for each class, each descriptor is represented with a different colour, for quantitative descriptors their mean value is shown, for qualitative descriptors each modality is represented with a different colour, and their occurrence frequency is shown (the sum of all modalities for a qualitative descriptor must be 1).
It is also possible to view the complete information for each class in text mode (**Class details** button). See Figure 22.

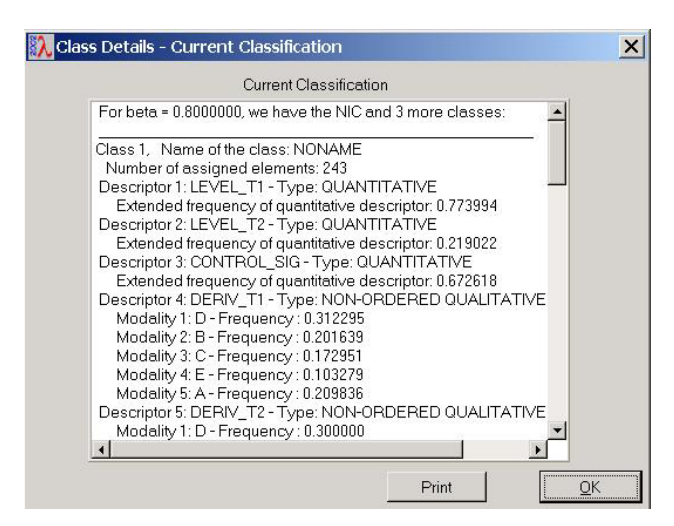


Figure 22 – Class Details window

The Class Details window shows all the parameters values for each class. Depending on the type of algorithm used for classification different information will appear. Also the number of elements (individuals) assigned to each class are shown.

- When Recognition is launched, and if the loaded population has pre-defined classes a 'Results Window' is displayed with different information (Figure 23): number (and percentage) of individuals not recognised, number of individuals with pre-defined classes, number (and percentage) of individuals badly recognised, and the list of the individuals badly recognised.

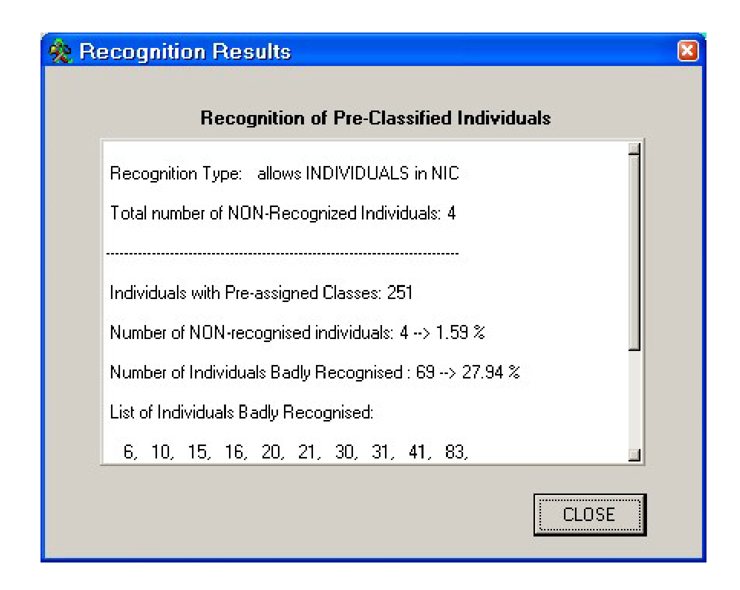


Figure 23 - Recognition Results Window

- When a population with pre-defined classes has been loaded the **"View Prof CL"** button is enabled. By clicking this button the pre-defined classes for each individual are displayed (**Figure** 24).

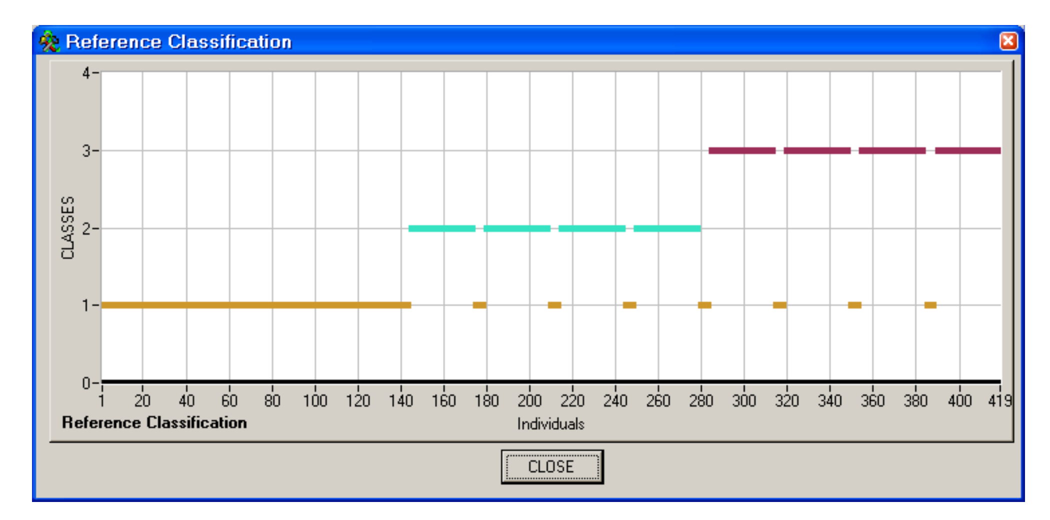
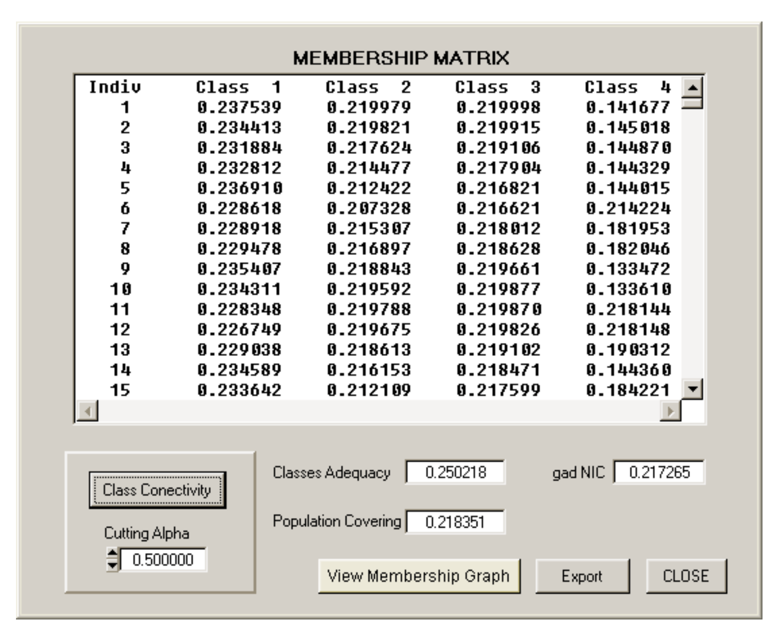


Figure 24 - Reference Classification Graph

- SALSA automatically creates a result file where each individual has its corresponding class. This file is a text file and has a .RES extension. It is possible to **save** the classification result as the 'reference' one by clicking on the **"Save Ref. CL"** button. The user may choose the name of the classification, and he can decide to set it as the reference for comparison with other classification results.

- The **ANALYSE** button shows the resulting membership matrix of each individual for each existing class (Figure 25). This matrix can also be graphically viewed by clicking on the **View Membership Graph** button (Figure 26). The user may save this matrix in a *.CSV format by clicking on the **EXPORT** button.


MEMBERSHIP MATRIX

Indiv	Class 1	Class 2	Class 3	Class 4
1	0.237539	0.219979	0.219998	0.141677
2	0.234413	0.219821	0.219915	0.145018
3	0.231884	0.217624	0.219106	0.144870
4	0.232812	0.214477	0.217904	0.144329
5	0.236910	0.212422	0.216821	0.144015
6	0.228618	0.207328	0.216621	0.214224
7	0.228918	0.215307	0.218012	0.181953
8	0.229478	0.216897	0.218628	0.182046
9	0.235407	0.218843	0.219661	0.133472
10	0.234311	0.219592	0.219877	0.133610
11	0.228348	0.219788	0.219870	0.218144
12	0.226749	0.219675	0.219826	0.218148
13	0.229038	0.218613	0.219102	0.190312
14	0.234589	0.216153	0.218471	0.144360
15	0.233642	0.212109	0.217599	0.184221

Figure 25 - Analyse Window

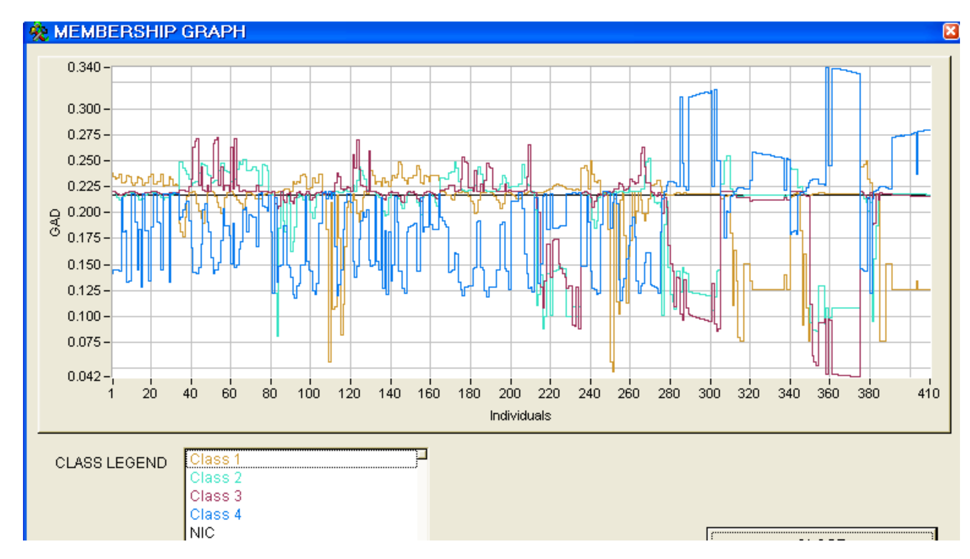


Figure 26 - Membership Graph

3.2.1.7 Mapping Classes to States

- When a suitable classification is obtained the user must assign the different classes into representative states. This is done using the **'Mapping to States'** button, located at the right bottom side of the 'Current Classification' graph.

- The user must first of all create the list of possible significant states, by clicking at the 'Create List of States' button. (See Figure 27).

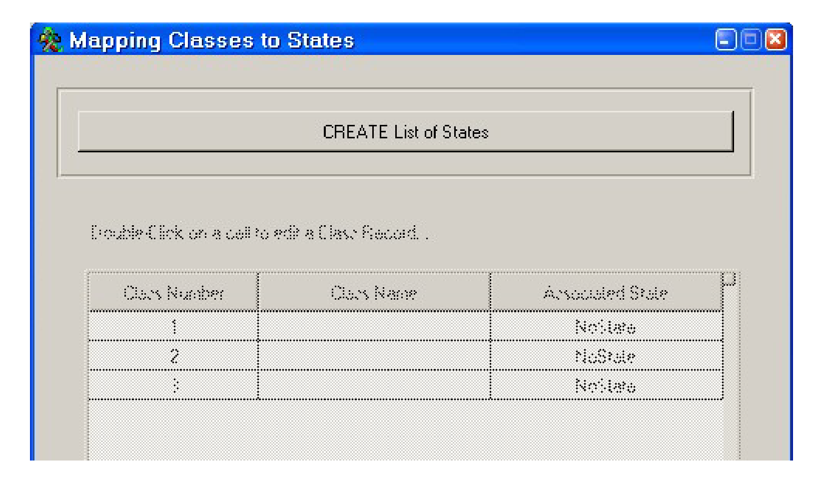


Figure 27 - Mapping Classes to States Window

- The user can add and remove the number of representative states of the process. Once the list is completed the user can close the window by clicking the OK button (See Figure 28).

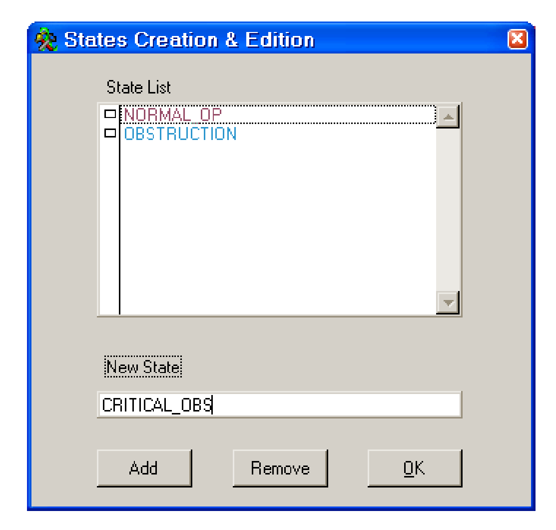


Figure 28 - Creating the list of States

- After the list of states is created the table of classes and states must be filled up. A class can be mapped into a state from the list by double-clicking into the cell of the corresponding class (Figure 29).

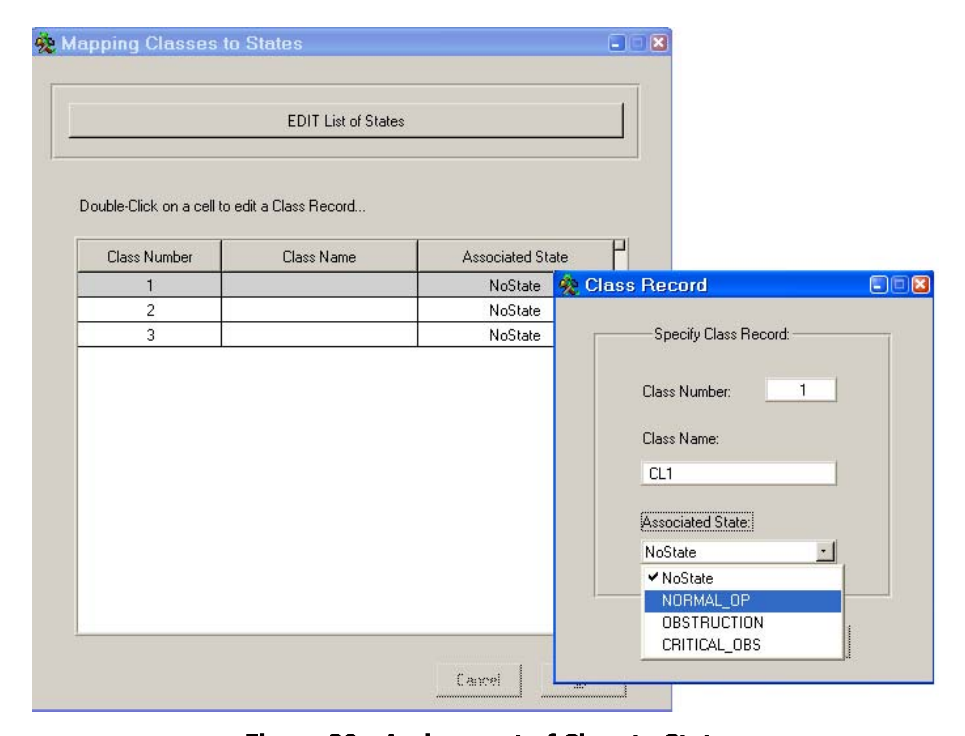


Figure 29 - Assignment of Class to State

- The table may be viewed or edited with the **"View/Edit Mapping"** button. The user can also view the assignment of states to the individuals by clicking the **States** button (Figure 30).



Figure 30 - View/Edit Mapping & States buttons

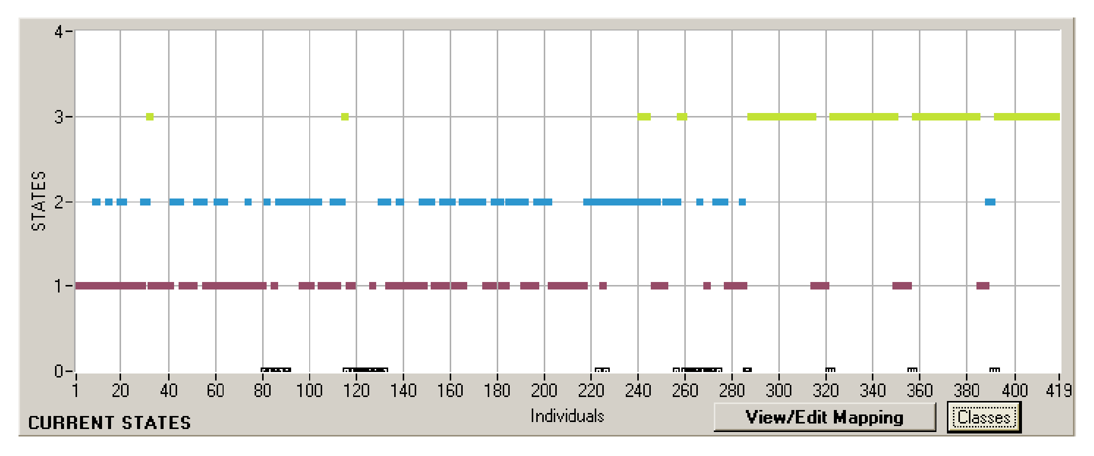


Figure 31 - Current States Graph

- SALSA allows saving the different contexts used (active descriptors, presence function for quantitative descriptors, connectives and exigency level), as well as the classification results for future use. Context will be saved in a binary format with a .CON extension. Whereas, a classification parameters (classes and states) will be saved with in a binary format with a .CLA extension (See Figure 18 for location of **save** buttons for context and classification).

3.3 SALSA – ONLINE

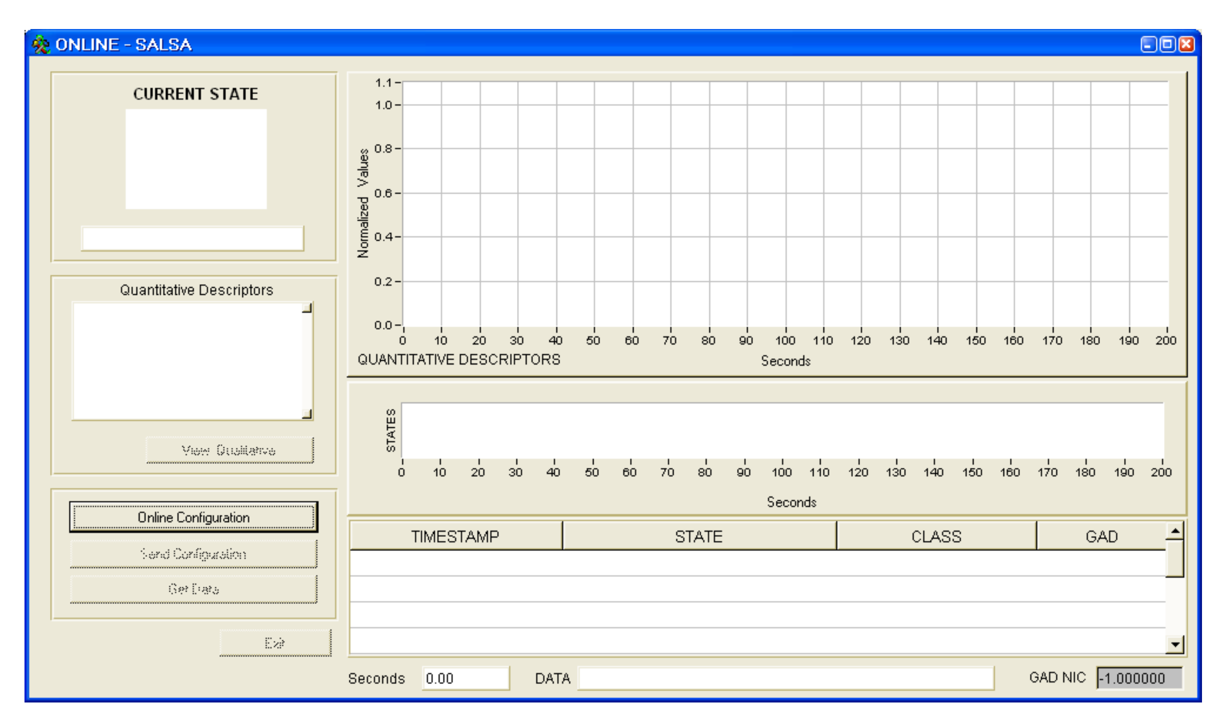


Figure 32 - SALSA Online Main Screen

These are the steps the user must follow for online recognition.

3.3.1 Online Configuration

The online configuration button is the only enabled button when the user chooses the online stage of SALSA in the main menu (**Figure** 32).

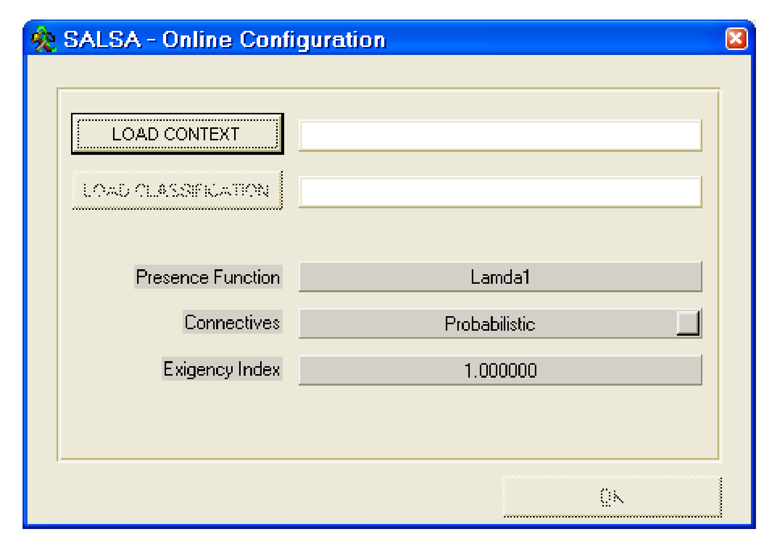


Figure 33 - SALSA Online Configuration

- The user must load up the context that will be used for recognition, i.e. the number and type of descriptors used on the offline stage for the creation of classes (**Figure** 33). This file is a binary file with a .CON extension (this file should have been previously created during the offline stage).
- Once the context is loaded the user must choose a classification file with .CLA extension. This classification must have the same number and type of descriptors as the loaded context. Once the user has loaded the classification, the learning parameters used for this classification will appear under the labels: "Presence Function", "Connectives", and "Exigency Index".
- Once both files have been loaded the user may close the configuration window by clicking on the **OK** button.

3.3.2 Online Connection

Once the configuration window is closed the user will see the list of the quantitative descriptors (or the qualitative descriptors) that will be received online. See Figure 34

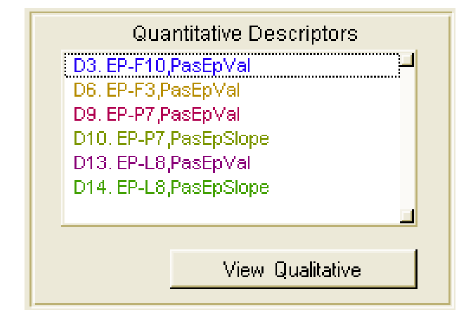


Figure 34 – List of Online Quantitative Descriptors

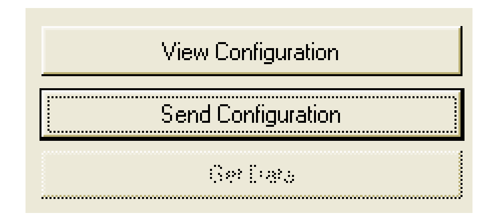


Figure 35 – Ready to Send Online configuration

The Send Configuration button is enabled (**Figure** 35). The user must click this button for SALSA to get ready to receive online data. If the connection is successful SALSA will start checking for a new message to arrive.

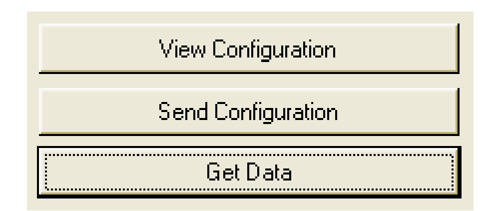


Figure 36 – Ready to Receive Online data

The **Get Data** button is enabled (Figure 36). SALSA will display the path where the data transfer will take place (Figure 37). The "DataTr" folder must contain the file with the current information message. This file has the name: **d_in.dat**. It contains one single line in the same format as for the training files (see Figure 38). Once SALSA reads the d_in.dat file it destroys it, and when recognition is made an output file is generated (**d_out.dat** see Figure 39) the application that is sending the online data must destroy this file and create the d_in.dat file to tell SALSA that a new message has arrived.

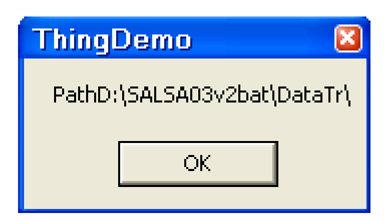


Figure 37 – Path for data transfer

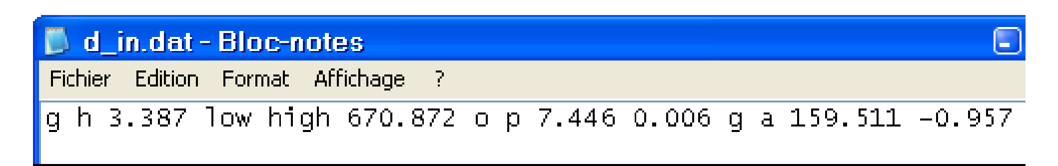


Figure 38 – Format for the online data file

d_out.dat - Bloc-notes
Fichier Edition Format Affichage ?
02-06-2004 - 11:39:49 NORMAL NoRegul 1.12262e-05

Figure 39 – Format for the output file by SALSA

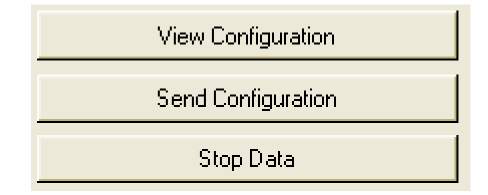


Figure 40 – Stop receiving data

The user may stop receiving messages with the **Stop Data** button.

3.3.3 Receiving Online Data

The online screen offers the visualization of the incoming data, the current state, the state of the plant at the instance of an arriving message it has a historical table with the time at which there was a change of state, with its associated class and Global Adequacy Degree.

3.3.3.1 Visualizing incoming data

The user may choose to visualise either the quantitative or the qualitative data.

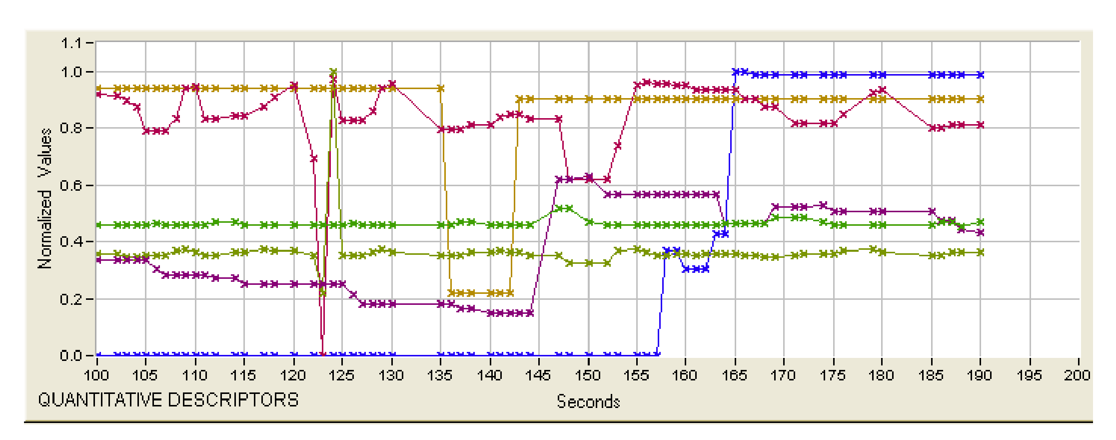


Figure 41 - Online Quantitative Descriptors

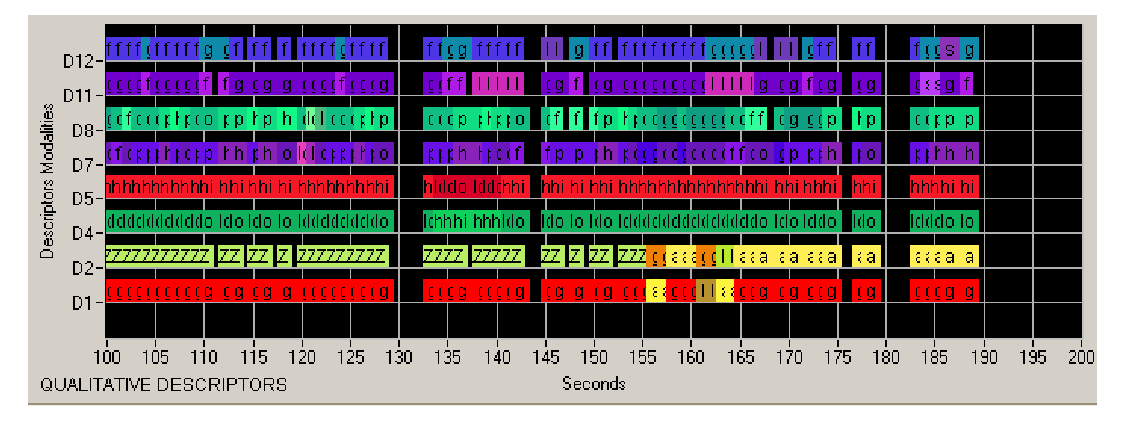


Figure 42 - Online Qualitative Descriptors

3.3.3.2 Visualizing the Current State

- The user may visualise what the current state of the plant is, with the rectangle at the top left corner of the screen:

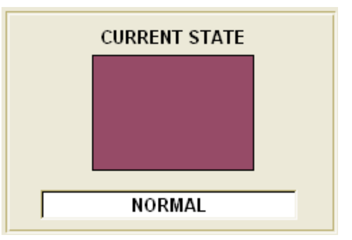


Figure 43 - Current State

- A bar graph with the state of the plant each time a new message data arrives is also available:

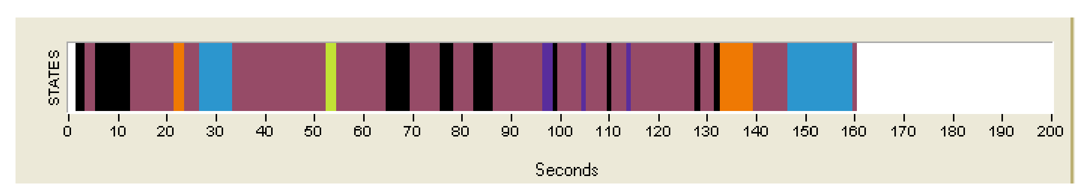


Figure 44 - Bar Graph with States

- A Table with the time of the change of state, the state, its associated class and the membership of the message to that class for the last 10 states is shown. The current state is highlighted with the corresponding colour of the state:

TIMESTAMP	STATE	CLASS	GAD
10-17-2003 - 17:46:05	NORMAL	NoRegul	1.00006e-05
10-17-2003 - 17:46:03	B_LEAKAGE	BoilerLeak	9.99993e-06
10-17-2003 - 17:45:49	NORMAL	NoRegul	1.0001e-05
10-17-2003 - 17:45:48	STABLE	BoilLevelStable	1.0003e-05

Figure 45 - Table with change of State occurrences

To EXIT the online screen the user must **FIRST STOP** the online connection with the **STOP** button and then **EXIT** the screen with the EXIT button.

4. APPENDIX

4.1 ONLINE DATA GENERATOR

This application permits to generate online data from stored data files. It can be used in order to validate SALSA Online stage.

The Online Data Generator is a standalone executable that may be installed in the following way:

1. In the InstallationSALSA04v1 Folder look for the sub-folder: Package. Click the 'setup.exe' file.

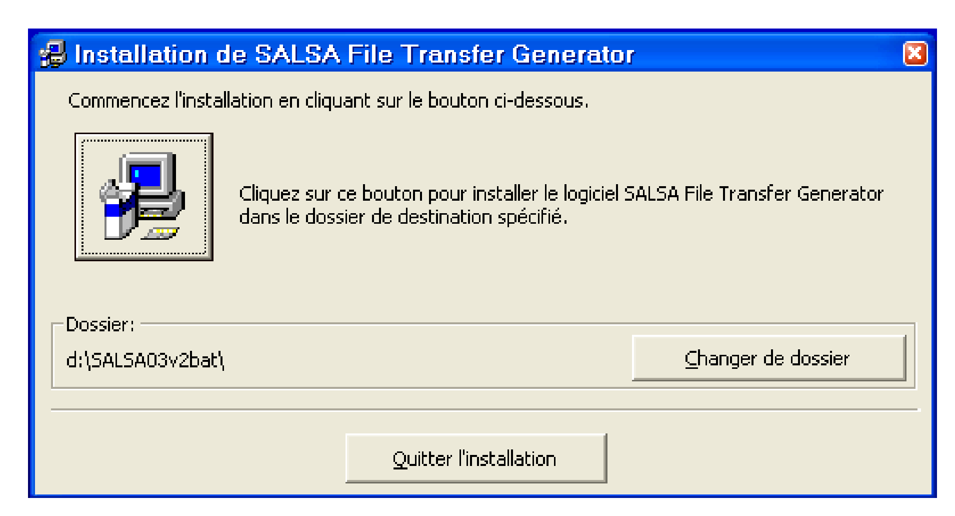


Figure 46 - Online Data Generator setup window

2. Once the installation is completed it is possible to generate online data from a file.

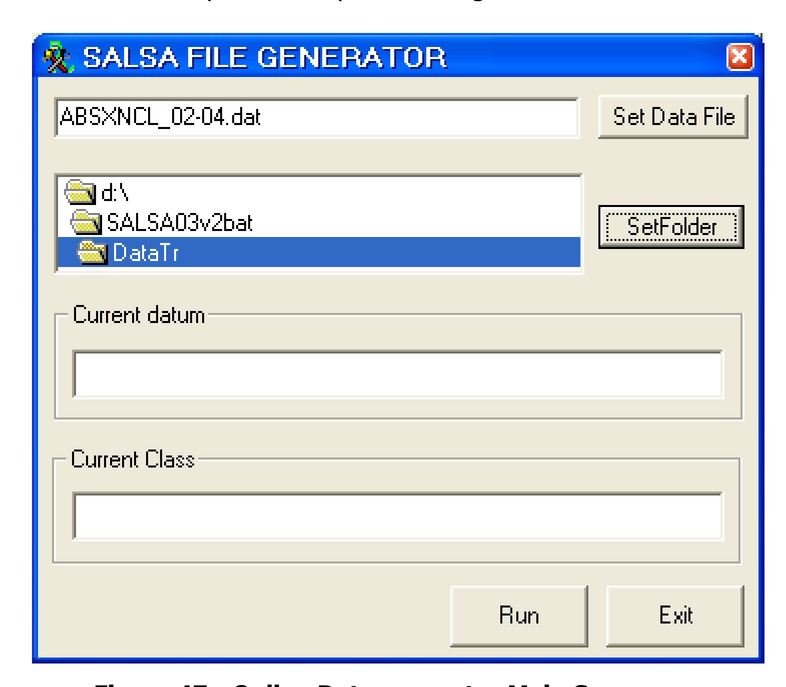


Figure 47 - Online Data generator Main Screen

3. First of all choose the data file with extension *.DAT and click the **Set Data File** button (see Figure 47).
4. Choose the folder where the online data file must be created (it must be in the \DataTr folder created by SALSA: **...\SALSA04v1\DataTr)** and click the **Set Folder** button
5. The RUN button is enabled. When clicking this button the program will create the **d_in.dat** file, the information of this file will be displayed in the 'current datum' window. Online Data is generated every two seconds.
6. When SALSA creates the output file **d_out.dat** the program will display its contents in the 'Current Class' window, and will destroy this file in order to generate a new d_in.dat file with a new datum.
7. To quit the application click on the EXIT button.

5. REFERENCES

[Kempowsky, Aguilar-Martin, Le Lann, Subias] *Learning Methodology for a Supervision System using LAMDA Classification Method.* IBERAMIA'02 - VIII Iberoamerican Conference on Artificial Intelligence, Sevilla (Spain), Nov. 12-15 2002

[Kempowsky, Aguilar-Martin, Subias, Le Lann] *Classification Tool based on Interactivity between Expertise and Self-Learning Techniques*. IFAC-Safeprocess 2003, Washington D.C. (USA), June 9-11 2003